ABC INFORMATIQUE

N°50

COURS D'INFORMATIQUE PRATIQUE ET FAMILIALE

QL Sinclair, un bluff ?

Jeu du Pendu

LOGO, première

Interface homme-machine

EDITIONS ATLAS

M 6062-50-12 F 85 FB - 3,80 FS - $ 1.95

Page manquante
(publicité et colophon)

Un pas en avant

Clive Sinclair bluffe-t-il à nouveau? Il compare son dernier ordinateur, le QL, à des machines coûtant cinq fois plus cher, comme l'IBM PC ou le Macintosh de chez Apple!

La famille Sinclair
Le label Sinclair figure sur une vaste gamme de matériels très divers, allant des amplificateurs audio jusqu'aux calculateurs et ordinateurs. Il y a également des chimères Sinclair, matériels qui se font attendre, comme une voiture électrique et un écran de télévision ultra-plat. L'innovation technologique est toujours présente — avec un design très recherché, style « hi-tech ». A ses atouts de modernisme s'est ajoutée une stratégie de marketing ambitieuse. Les esprits critiques diront que ce qui caractérise plutôt Sinclair, c'est une « ingénierie » des communiqués de presse, l'étalage de gadgets et des délais particulièrement longs de lancement des produits et de livraison. (Cl. Ian McKinnell.)

Que penser du Sinclair QL? La publicité massive et agressive de la période du lancement annonçait l'ordinateur personnel le plus performant du marché, s'appuyant sur un microprocesseur de 32 bits, le 68008, un ensemble de logiciels intégrés (dont quatre logiciels de gestion), et un BASIC résident devant surpasser tout ce qui se faisait alors.

Très vite, les délais de livraison ne furent plus tenus, et l'on commença à se dire que les performances annoncées ne cadraient pas avec la réalité. La crédibilité de Sinclair reçut un coup supplémentaire, lorsqu'on réalisa qu'en dépit des assurances du constructeur, la machine était loin d'être prête. Et des clients payaient d'avance!

Pas de doute : l'opération ne ressemblait pas à la sortie du précédent micro, le ZX Spectrum. Même si pour ce produit les délais de 28 jours ne furent pas respectés, les journalistes comme les spécialistes avaient pu disposer de machines de test dès l'annonce du lancement. Et les essais avaient alors confirmé les performances annoncées. Pour le QL, Sinclair dut reconnaître que le « SuperBASIC » clamé et le système d'exploitation ne pourraient tenir ensemble dans les 32 K de ROM alloués. Il fallait encore rajouter 48 K, mais il n'y avait pas de place sur la carte de circuits imprimés pour le composant supplémentaire!

Plutôt que de perdre encore du temps et de l'argent à repenser la carte des circuits imprimés, l'équipe de Sinclair s'en tira en rajoutant, à l'extérieur du port de la cartouche mémoire du QL, une sorte de protubérance en plastique noir contenant la partie manquante du BASIC et du système d'exploitation!

Cela permit de sortir des machines utilisables, mais les vingt-huit jours de délai de livraison prévus étaient devenus trois mois! Et plusieurs versions du système d'exploitation allaient suivre, chacune avec ses propres défauts... Sinclair finit par se décider pour une version appelée « AH » qui fut la première à être commercialisée en masse.

Les réactions de l'utilisateur à la « boîte de Sinclair » furent très mauvaises. Cet appendice était la preuve visible de la défaillance de la machine. Quant à sa fiabilité : l'échec. En outre, Sinclair

dut recourir à un bricolage sur la carte elle-même. En effet, il n'y avait pas de support sur la carte pour le composant supplémentaire. Le troisième composant de 16 K de ROM, qui est en surnombre, a donc été placé par-dessus un autre composant. Sur vingt-huit broches du composant « parasite », vingt-sept ont été directement soudées à autant de broches correspondantes de son « hôte ». La dernière a été connectée, par un fil « volant », à une autre partie de la carte, afin que le nouveau composant soit indépendant. Grâce à ce « bidouillage », la boîte d'extension fut bien supprimée, mais sans remédier au problème de fond.

Les machines antérieures utilisaient des mémoires mortes reprogrammables électriquement (EPROM) à la place de ROM. Sinclair pouvait ainsi être en avance sur la production des ROM. En contrepartie, cela revenait très cher — ces composants de EPROM de 16 K se vendaient environ 100 F l'unité. Même si Sinclair bénéficiait de rabais, cela devait représenter une part importante du coût de la machine. C'est peut-être la raison pour laquelle Sinclair a retenu de manière hâtive la version AH du système d'exploitation, malgré ses défauts. Une fois installés, de nouveaux composants de ROM pouvaient venir remplacer les coûteux composants de mémoire morte reprogrammable. La rentabilité de la machine était alors à ce prix. Toujours est-il qu'il faudra attendre 1985 pour avoir une version sans défaut de la version AH.

Après tout ce discours, comment apprécier le QL ? Il s'agit de deux machines en une seule : un micro personnel puissant, et un petit ordinateur de gestion. En cela, il est vraiment novateur. Néanmoins, le QL est davantage un micro-ordinateur conventionnel. Il est petit, peut générer un affichage télé, il comporte un BASIC résident, dispose d'un mode graphique couleur haute résolution et de deux ports manette de jeu. Pourtant, deux de ses caractéristiques lui permettraient de s'apparenter aux micros de gestion : ses microlecteurs incorporés, qui autorisent une sauvegarde de masse substantielle (en tous cas, par comparaison avec une cassette), et quatre programmes d'application, livrés avec lui (un traitement de texte, un tableur, une base de données et un logiciel graphique).

En tant qu'ordinateur individuel, le QL est très intéressant pour ses microlecteurs et ses logiciels. Les logiciels d'accompagnement sont de bonne qualité et permettent d'accéder à des programmes réservés jusque-là à des spécialistes. Mais reconnaissons que la plupart des utilisateurs de micro-ordinateurs personnels n'ont jamais besoin de ces programmes. L'ordinateur domestique est surtout destiné à des jeux et à permettre l'écriture de programmes en BASIC. Le SuperBASIC est tout à fait indiqué, pour sa part, pour le développement de programmes BASIC, dans la mesure où il s'agit probablement du meilleur BASIC créé jusqu'à présent. Toutefois, les logiciels disponibles pour le QL sont encore rares. La commercialisation de logiciels pour le QL est en outre entravée par l'incompatibilité entre les nouveaux microlecteurs et le Spectrum, par un dessin d'écran différent de celui du Spectrum, et par les particularités système d'exploitation.

Il y a semble-t-il toujours des défauts dans le SuperBASIC, et l'éditeur est très semblable à l'éditeur de lignes utilisé par le Spectrum (ce qui n'est pas un défaut, bien sûr, mais cela n'a rien à voir avec les standards de la 5e génération). L'ajout de procédures et de fonctions à la manière du BBC Micro, et d'une structure SELECT semblable à CASE en PASCAL, constituent de réelles améliorations, bien que la commande ON ERROR... ne soit pas développée. Les capacités graphiques sont très bonnes et le son est correct, mais tout cela

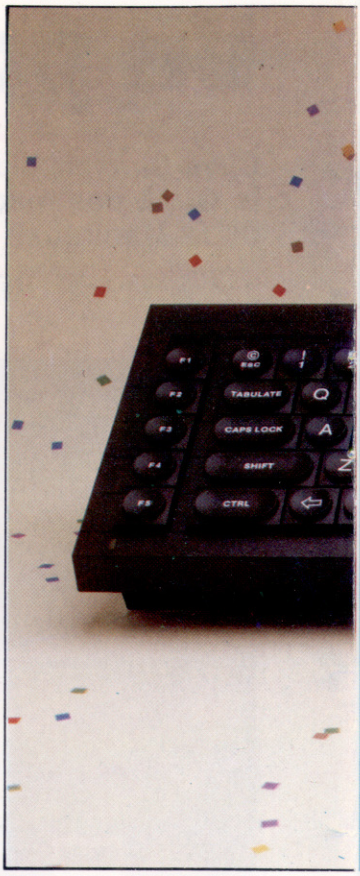
En compétition

QL contre BBC Micro

Pour prétendre à l'important marché des micro-ordinateurs personnels, le QL doit pouvoir rivaliser avec le BBC Micro. En faveur du QL, la mémoire libre, la mémoire de masse, un microprocesseur innovateur, un graphisme performant de grande capacité mémoire susceptible d'être augmentée. En sa défaveur, le QL ne dispose pas d'un immense catalogue de logiciels, ni d'une vaste gamme de périphériques de haute qualité, ni du recours possible à un deuxième processeur ou à une extension.

QL contre Macintosh

Bien que les deux machines partagent certaines caractéristiques, dont la taille mémoire, la rapidité de l'horloge de l'UC et le principe du logiciel intégré, il n'y a pas de commune mesure entre le QL et le Macintosh. Le QL est techniquement une machine brillante, mais n'ayant rien de vraiment nouveau dans son mode de fonctionnement. Le Macintosh d'Apple dispose d'un système d'exploitation avec des caractéristiques entièrement originales, telles que figures graphiques et fenêtres à l'écran, et surtout la fameuse « souris », qui distingue de manière catégorique la machine de toutes celles présentes sur le marché. Le Macintosh représente bien ce « bond en avant » *(quantum leap)* qu'était censé être le QL, le premier maillon d'une nouvelle génération de micro-ordinateurs.



Logiciels intégrés

Les quatre logiciels présentent des formats et des commandes similaires à l'écran, ainsi qu'une représentation claire et concise. Les données peuvent être échangées par l'intermédiaire du lecteur de disque. Quill est un traitement de texte permettant l'affichage de 40, 64 ou 80 caractères. Abacus est un tableur disposant de nombreuses fonctions. Archive est une base de données possédant de nombreuses commandes pour des fichiers simples. Easel est un logiciel de création graphique.

Ian McKinnell

Sinclair QL

PRIX

***

DIMENSIONS

472 × 138 × 46 mm.

UNITÉ CENTRALE

Motorola 68008 — 7,5 MHz.

MÉMOIRE

128 K RAM (extension jusqu'à 640 K).
48 K ROM.

ÉCRAN

25 lignes de 85 caractères (avec moniteur); haute résolution graphique :
512 × 256 pixels (4 couleurs),
256 × 256 pixels (8 couleurs).

INTERFACES

Série RS232 (2), manettes de jeu (2), microlecteurs, TV, moniteur RVB.

LANGAGE DISPONIBLE

SuperBASIC

CLAVIER

Type pseudo-machine à écrire; 65 touches dont une vraie barre d'espacement et 5 touches de fonction.

DOCUMENTATION

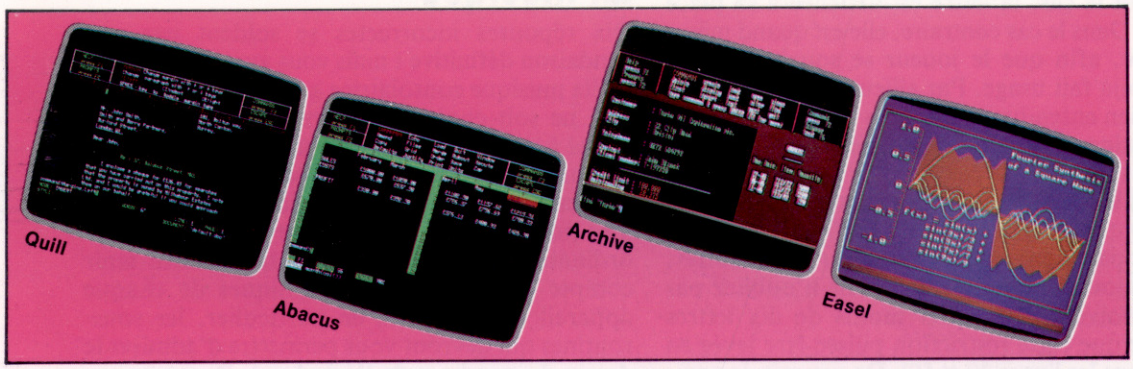
Le manuel utilisateur (en classeur) est de bonne qualité.

ATOUTS

Unité centrale 68008 très rapide, logiciels intégrés de haute qualité, graphique excellent, langage basic très performant.

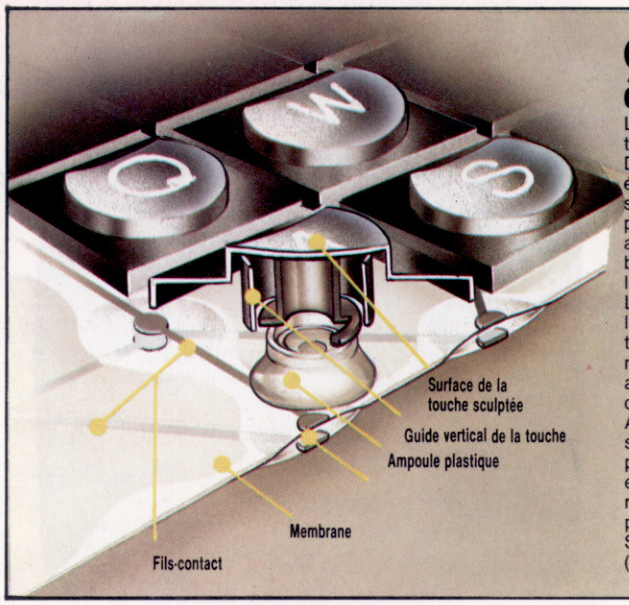
POINTS FAIBLES

Les microlecteurs incorporés sont plus lents que des lecteurs de disques, pas de compatibilité avec le Spectrum, peu de logiciels disponibles, système d'exploitation non exempt de défauts.



Ian McKinnell

est loin de ce que Sinclair laissait espérer. En tant que machine de gestion, le QL est encore moins convaincant. Le logiciel intégré est intéressant, bien que tout utilisateur gestionnaire veuille sûrement en plus un logiciel de comptabilité. Le programme du tableur, quant à lui, ne laisse que 15 K de mémoire utilisateur, ce qui exclut tout recours à un logiciel de simulation financière. La vitesse et la fiabilité douteuse des micro-lecteurs mettent en question la capacité de sauvegarde de mémoire de masse dans son ensemble (il n'existe pas d'interface lecteur de disque). Le clavier ne semble pas pouvoir s'accommoder d'un usage intensif quotidien, et on imagine mal des dactylos confirmées accepter de se plier à ses nombreuses originalités. Le manque de logiciels est un handicap commercial encore plus grand pour l'utilisateur gestionnaire éventuel. Si on ajoute à cela le manque de mémoire de masse, on est tenté de mettre rapidement (trop peut-être) un terme à la carrière professionnelle du QL.



Clavier à bulle

Le QL a un clavier de type « membrane ». Deux fils destinés à émettre un signal sont séparés par une bulle en plastique. Lorsque l'on appuie sur la touche, la bulle est comprimée et le contact s'établit. La résistance à l'enfoncement de la touche ainsi que son retour sont donc assurés par cette sorte d'ampoule de plastique. Avec ses touches sculptées à course pleine, et son profil bas et plat, le clavier du QL représente un gros progrès sur celui du Spectrum.
(Cl. Rosalind Buckland.)

Kevin Jones

Au gré des vents

Le jeu du « Pendu » offre l'avantage d'être facilement mis en œuvre sur micro-ordinateur. Sa programmation permet, de surcroît, d'approfondir la manipulation des chaînes de caractères.

Tout le monde a déjà joué au Pendu au moins une fois dans sa vie ! Il s'agit simplement de deviner un mot, dont on ne connaît au départ que le nombre de lettres (représentées par des tirets). Le joueur propose chaque fois une lettre. S'il a vu juste, la lettre correspondante est affichée à sa place exacte. S'il s'est trompé, l'image d'un pendu accroché à sa potence s'enrichit d'un élément. Dans notre programme, cette image comporte dix parties ; elle apparaît à droite de l'écran. Vous perdrez si elle est complétée avant que vous ayez eu le temps de donner toutes les lettres du mot.

Le principe de base de notre programme est des plus simples. Il consiste à s'assurer qu'une lettre tapée au clavier est bien contenue dans un des mots « secrets » choisi au hasard parmi les onze qui sont contenus dans les DATA, qu'on lira à la fin des programmes développés plus loin. Si tel est le cas, la lettre est affichée à sa place exacte. Dans le cas contraire, elle est quand même affichée, afin que le joueur se souvienne d'en avoir déjà fait usage. Par ailleurs, le programme doit, bien entendu, passer à un sous-programme qui dessine sur l'écran un élément de l'image du pendu.

Dans les deux versions du programme, les mots sont contenus dans les lignes 1020 et 1030. Vous pouvez parfaitement les remplacer par d'autres, ou en ajouter de nouveaux, mais n'oubliez pas qu'ils doivent comporter moins de dix lettres (encore que cette restriction puisse être levée en modifiant les lignes 30 et 50). De surcroît, le nombre total de mots contenus dans les DATA doit être le même que celui utilisé à la ligne 20.

Au début du programme, tous les mots sont placés dans un tableau ; l'un d'entre eux est sélectionné au hasard, et une ligne de tirets est affichée à l'écran. Il y a autant de tirets que de lettres. Le reste du jeu se réduit à une boucle sans cesse répétée. Chaque lettre tapée au clavier est examinée ; s'il y a plus d'une lettre, ou si ce n'est pas un caractère, un « bip » se fait entendre, et le programme attend qu'on lui propose une réponse acceptable. La lettre est aussi comparée à celles qui ont déjà été utilisées, et un message d'avertissement sanctionne tout emploi répété.

Si ces deux procédures de vérification donnent des résultats positifs, la lettre est ajoutée aux caractères déjà mis en œuvre, puis comparée au mot recherché, lettre par lettre. En cas de concordance, elle est placée au bon endroit. Sinon, le sous-programme se chargera de compléter l'image du pendu.

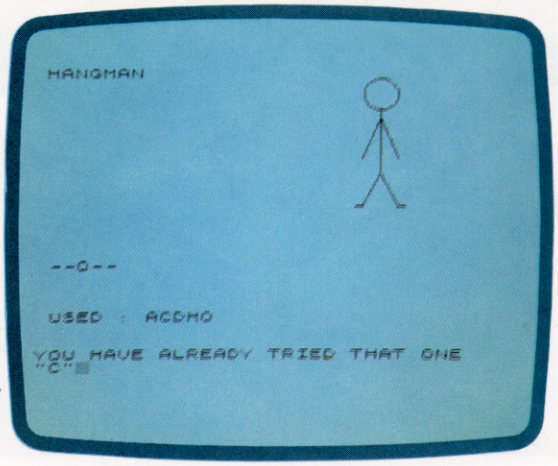
En musique

Dix réponses incorrectes se soldent par une pendaison définitive : mais vous aurez droit à une petite musique consolatrice avant qu'un nouveau mot ne soit choisi. Si, en revanche, vous avez réussi à deviner toutes les lettres du mot, un air triomphal accompagnera votre victoire.

Il est facile de transposer nos deux programmes pour les faire tourner sur d'autres micro-ordinateurs, mais il faudra naturellement tenir compte des possibilités graphiques de chaque appareil en adoptant, en particulier, les sous-routines qui devront vous permettre d'admirer le dessin du pendu et de l'échafaud. Rien ne vous empêche, à ce sujet, de créer un dessin très élaboré : que diriez-vous par exemple d'un pendu se balançant doucement au gré des vents à chaque fois qu'un joueur se trompe ?

Le jeu est susceptible d'être amélioré de bien des façons. Il serait par exemple très utile d'ajouter en début de programme une procédure de vérification qui empêcherait tout mot déjà sélectionné d'être choisi une fois de plus. Ici, en effet, il est simplement déterminé de façon purement aléatoire, et rien ne s'oppose à ce qu'il soit tiré deux fois de suite.

Il serait encore plus utile d'élaborer une routine vérifiant que seules les lettres majuscules seront acceptées (ce qui n'est pas le cas ici). Minuscules, nombres et caractères seraient rejetés après détermination de leur code ASCII. Mais, tel qu'il est, notre programme vous permettra de vous offrir une bonne partie de Pendu, tout en vous laissant de nombreuses possibilités d'amélioration.



Liz Heaney

Évolution
Le développement de la forme du pendu et la progression du mot se font en parallèle.

BBC Micro

```
10 REM Jeu du pendu
20 N=11
30 DIM W$(N)
50 D$="----------"
60 FOR I=1 TO N
70 READ W$
80 W$(I)=W$
90 NEXT I
100 MODE 1
110 REM Début d'un nouveau mot
120 CLS
130 PRINT TAB(5,1); "PENDU !"
140 W$=W$(RND(N))
150 C=0 : W=0 : D=0
160 G$=""
170 PRINT TAB(5,23); "Lettres utilisées :"
180 L=LEN(W$)
190 PRINT TAB(5,20);LEFT$(D$,L)
200 REM La lettre proposée est vérifiée
210 PRINT TAB(5,30);STRING$(34," ")
220 INPUT TAB(5,30);L$
230 REM Vérifier lettre entrée
240 IF L$="" THEN SOUND 1,-15,50,5: GOTO 210
250 IF LEN(L$)>1 THEN SOUND 1,-15,50,5: GOTO 210
260 F=0
270 FOR I=1 TO LEN(G$)
280 IF L$<>MID$(G$,I,1) THEN GOTO 330
290 PRINT TAB(5,30)"LETTRE DEJA PROPOSEE !"
300 SOUND 1,-15,50,5
310 TX=TIME : REPEAT UNTIL TIME>TX+50
320 F=1
330 NEXT I
340 IF F=1 THEN 210
350 G$=G$+L$
360 PRINT TAB(5,25);G$
370 REM Comparaison de la lettre et du mot
380 F=0
390 FOR I=1 TO L
400 IF L$=MID$(W$,I,1) THEN GOSUB 480
410 NEXT I
420 IF F<>1 THEN GOSUB 670
430 IF D=1 THEN GOSUB 590 : GOTO 120
440 IF C=L THEN GOSUB 530 : GOTO 120
450 GOTO 210
460 END
470 REM La lettre correspond
480 PRINT TAB(4+I,20);L$
490 C=C+1
500 F=1
510 SOUND 1,-15,200,5 : SOUND 1,0,0,2
520 RETURN
530 REM Gagné !
540 FOR I=1 TO 200 STEP 10
550 SOUND 1,-15,I,2
560 NEXT I
570 RETURN
580 REM Perdu !
590 FOR I=200 TO 1 STEP -10
600 SOUND 1,-15, I,2
610 NEXT I
620 CLS
630 PRINT TAB(5,20); "LE MOT ETAIT":W$
640 TX=TIME:REPEAT UNTIL TIME>TX+200
650 RETURN
660 REM La lettre ne correspond pas
670 W=W+1
680 ON W GOTO 690,760,780,810,830,870,890,910,930,960
690 VDU29, 800;800;
700 MOVE 50,0
710 FOR A=0 TO 7 STEP 0.4
720 DRAW 50*COS(A),50*SIN(A)
730 NEXT A
740 VDU29,0;0;
750 RETURN
760 MOVE 800,750 :DRAW 800, 550
770 RETURN
780 MOVE 740,450 : DRAW 750,450 : DRAW 800,550
790 DRAW 850,450 : DRAW 860,450
800 RETURN
810 MOVE 750,630 : DRAW 800,730 : DRAW 850,630
820 RETURN
830 PLOT 69,800,800
840 PLOT 69,820,820
850 PLOT 69,780,820
860 RETURN
870 MOVE 780,790 : DRAW 800,770 : DRAW 820,790
880 RETURN
890 MOVE 600,400 : DRAW 1100,400
900 RETURN
910 MOVE 1000,400 : DRAW 1000,900
920 RETURN
930 MOVE 1000,900 : DRAW 800,900
940 MOVE 1000,850 : DRAW 950,900
950 RETURN
960 MOVE 800,900 : DRAW 800,850
970 MOVE 780,790 : PLOT 14,800,770 : PLOT 6,
820,790
990 D=1
1000 RETURN
1010 REM les mots
1020 DATA "MUTATION","DEBAT","GIRAFE","PENDU",
"BBC",
1030 DATA "SPECTRUM","GEOLE","ELEPHANT",
"XENOPHOBE","TRINGLE","GRENADE"
```

Spectrum

```
10 REM Jeu du Pendu
20 LET N=11
30 DIM X$(N,10)
35 DIM Y(N)
40 REM Initialisation
50 LET D$="----------"
60 FOR I=1 TO N
70 READ W$
75 LET Y(I)=LEN W$
80 LET X$(I)=W$
90 NEXT I
110 REM Début d'un nouveau mot
120 CLS
130 PRINT AT 1,1;"PENDU !"
140 LET I=(1-INT (RND*N))
145 LET W$=X$(I)
150 LET C=0 : LET W=0 : LET D=0
160 LET G$=""
170 PRINT AT 20,1 ; "LETTRES EMPLOYEES : "
180 LET L=Y(I)
190 PRINT AT 16,1; D$(1 TO L)
200 REM Entrée d'une lettre
220 INPUT L$
230 REM Vérification de la lettre
240 IF L$="" THEN BEEP 0.25,-10 : GO TO 210
250 IF LEN L$)1 THEN BEEP 0.25,-10 : GO TO 210
260 LET F=0
270 FOR I=1 TO LEN G$
280 IF L$<>G$(I) THEN GO TO 330
290 PRINT AT 21,1; "LETTRE DEJA EMPLOYEE !"
300 BEEP 0.25,-10
310 PAUSE 25 : PRINT AT 21,1,,
320 LET F=1
330 NEXT I
340 IF F=1 THEN GO TO 210
350 LET G$=G$+L$
360 PRINT AT 20,B10$
370 REM Compare lettre et mot
380 F=0
390 FOR I=1 TO L
400 IF L$=W$(I) THEN GO SUB 480
410 NEXT I
420 IF F<>1 THEN GO SUB 670
430 IF D=1 THEN GO SUB 590 : GO TO 120
440 IF C=L THEN GO SUB 530 : GO TO 120
450 GO TO 210
460 STOP
470 REM La lettre concorde
480 PRINT AT 16,I;L$
490 LET C=C+1
500 LET F=1
510 BEEP 0.25,+16
520 RETURN
530 REM Gagné !
540 FOR I=-10 TO 10
550 BEEP 0.1,I
560 NEXT I
570 RETURN
580 REM Perdu !
590 FOR I=10 TO -10 STEP -1
600 BEEP 0.1,I
610 NEXT I
620 CLS
630 PRINT AT 10,3;"LE MOT ETAIT : "!W$
640 PAUSE 50
650 RETURN
660 REM La lettre ne concorde pas
670 LET W=W+1
680 IF W=1 THEN GO TO 690
681 IF W=2 THEN GO TO 760
682 IF W=3 THEN GO TO 780
683 IF W=4 THEN GO TO 810
684 IF W=5 THEN GO TO 830
685 IF W=6 THEN GO TO 870
686 IF W=7 THEN GO TO 890
687 IF W=8 THEN GO TO 910
688 IF W=9 THEN GO TO 930
689 IF W=10 THEN GO TO 960
690 CIRCLE 200,150,10
700 RETURN
760 PLOT 200,140 : DRAW, 0,-40
770 RETURN
780 DRAW 10,-20 : DRAW -4,0
790 PLOT 200,100 : DRAW 10,-20 : DRAW 4,0
800 RETURN
810 PLOT 190,110 : DRAW 10,25 : DRAW 10,-25
820 RETURN
830 PLOT 200,150
840 PLOT 196,155
850 PLOT 204,155
860 RETURN
870 PLOT 196,148 : DRAW 4,-4 : DRAW 4,4
880 RETURN
890 PLOT 170,60 : DRAW 80,0
900 RETURN
910 PLOT 240,60 : DRAW 0,115
920 RETURN
930 DRAW -40,0
940 PLOT 240,165 : DRAW -10,10
950 RETURN
960 PLOT 200,175 : DRAW 0,-15
970 OVER 1: PLOT 196,148 : DRAW 4,-4 : DRAW 4,4 :
OVER 0
980 PLOT 196,144 : DRAW 4,4 : DRAW 4,-4
990 LET D=1
1000 RETURN
1010 REM Les mots
1020 DATA "MUTATION","DEBAT","GIRAFE","PENDU",
"BBC"
1030 DATA "SPECTRUM","GEOLE","ELEPHANT",
"XENOPHOBE","TRINGLE","GRENADE"
```

Problème de taille

Notez que vous pouvez introduire dans ce jeu l'alternative entre lettres majuscules ou minuscules, de façon à ce qu'elles correspondent à la « nature » du mot mystérieux.

Le langage logo

LOGO est un langage de programmation conçu initialement dans un but éducatif. Quel est son principe, son histoire et le type d'utilisateurs qu'il peut intéresser?

Après avoir étudié de façon détaillée la programmation en BASIC et en code machine, nous nous tournons maintenant vers d'autres langages informatiques, en commençant par le LOGO. Pourquoi choisir ce langage? Il y en a beaucoup d'autres qui exécutent très bien certaines fonctions, et les ordinateurs domestiques offrent rarement le langage LOGO.

D'abord, LOGO est l'un des meilleurs langages d'introduction qui existent aujourd'hui sur le marché informatique. Évidemment, si vous savez déjà programmer en BASIC vous n'avez que faire d'un langage d'introduction. Mais, même pour le programmeur BASIC expérimenté, LOGO peut servir de bonne introduction à la programmation « structurée » et à l'utilisation de procédures au lieu d'instructions. Ensuite, LOGO est disponible sur cassette ou sur cartouche pour la plupart des ordinateurs domestiques. Finalement, LOGO représente un système d'enseignement très puissant. Bien que ce langage ne soit vraiment pas aisé à maîtriser, il est l'un des rares langages de programmation qui permettent des débuts assez faciles.

LOGO puise ses origines dans le langage d'intelligence artificielle LISP, qui fut inventé au début des années soixante pour permettre aux ordinateurs de gérer plus facilement des structures de données complexes. Son nom vient du fait que c'est un langage de « traitement de liste » *(list processing)*, ce qui signifie que sa structure de base de données est une liste, plutôt qu'une chaîne de caractères ou un tableau numérique, comme en BASIC. Les fonctions essentielles de LISP servent à manipuler les données à l'intérieur d'une liste. Les éléments peuvent être de simples symboles ou des listes entières. Résultat : les données non numériques (comme une phrase) sont traitées plus facilement de cette manière.

LISP repose avant tout sur le principe de la récurrence, où une chose (généralement une fonction ou une procédure) est définie par rapport à elle-même. Ces caractéristiques de LISP ne sont pas accidentelles, mais proviennent des recherches informatiques sur le langage naturel et sur l'intelligence humaine. Cependant, le langage n'est pas facile à apprendre, et, en 1968, un groupe de personnes associées au Massachussetts Institute of Technology (MIT) entreprit la conception d'un langage basé sur LISP, mais destiné aux enfants.

Ce groupe était dirigé par Seymour Papert, qui avait précédemment passé de nombreuses années à étudier avec Jean Piaget (1896-1980), sommité d'alors en psychologie éducative. En arrivant au MIT, Papert commença à travailler

Présentation de logo

LOGO a deux caractéristiques fondamentales qui font de lui un langage éducatif puissant. La première est sa nature interactive : lorsque vous tapez une commande, vous voyez immédiatement les résultats à l'écran. Cela signifie qu'il est facile de progresser (particulièrement pour les enfants et pour les débutants), parce que vous pouvez constater vous-même votre évolution.

La seconde caractéristique importante est la capacité d'extension de LOGO : des opérations complètes peuvent être gérées par des listes d'instructions LOGO élémentaires. Ces listes se nomment des procédures. Dès que l'une d'elles a été définie comme un ensemble d'instructions, le nom de cette procédure adopte le statut d'une nouvelle commande LOGO. A partir de là, la procédure entière peut être simplement exécutée en tapant son nom. De cette façon, vous pouvez créer vos propres commandes en plus des commandes « primitives » qui font partie du programme.

En programmation LOGO, la plupart des gens ont tendance à aller plus loin qu'avec d'autres langages. Ils commencent quelquefois avec un problème brut, écrivent une procédure pour le résoudre, et construisent ensuite un programme autour de cette procédure. La programmation en LOGO offre une approche beaucoup plus souple parce que, généralement, il existe plusieurs manières d'obtenir un résultat particulier.

Le père fondateur
Seymour Papert, le père de LOGO, est photographié ici à une conférence commanditée par Commodore en 1983. Papert est maintenant associé avec LOGO Computer Systems, qui fournit des programmes LOGO pour divers ordinateurs.

étroitement avec un expert en intelligence artificielle, Marvin Minsky.

Ces travaux sur LOGO se poursuivirent au cours des années soixante-dix, et d'autres groupes se formèrent pour expérimenter le nouveau langage. Le plus important de ceux-ci était basé à Édimbourg. Tout ce travail de développement fut mené dans des services de recherche universitaires, en utilisant des gros ou des mini-ordinateurs. Ce ne fut qu'avec l'arrivée des micro-ordinateurs que LOGO commença à être diffusé à une large échelle.

LOGO est un langage évolué qui nécessite beaucoup de mémoire, aussi bien pour le code que pour l'espace de travail. Les interpréteurs LOGO trouvés sur plusieurs micros nécessitent environ 30 K de mémoire, et encore 8 K ou plus pour l'affichage graphique. Tout cela avant même d'avoir commencé à programmer ! Bien qu'il eût été possible de mettre en œuvre des interpréteurs BASIC simples sur des micros domestiques dès leur mise sur le marché, ce n'est que lorsque des ordinateurs domestiques de plus de 48 K devinrent largement diffusés qu'il fut possible de proposer LOGO de manière intéressante.

Mais la publication du livre *Jaillissement de l'esprit,* de Seymour Papert, permit de sortir LOGO des services universitaires et de le faire connaître à un plus grand nombre de personnes. Dans son livre, Papert développe une théorie relative à l'utilisation éventuelle des ordinateurs dans l'éducation. Celle-ci était fondée sur trois éléments : la théorie de l'apprentissage, l'intelligence artificielle et le mouvement dans le monde éducatif cherchant à recentrer l'éducation autour de l'enfant. Papert désire voir les enfants programmer des ordinateurs, pour éviter le contraire (ce qui arrive le plus souvent en « éducation assistée par ordinateur »).

Le livre prévoit l'émergence d'une nouvelle « culture de l'ordinateur », dans laquelle certains concepts, jugés jusqu'ici hors des possibilités des enfants, pourraient facilement être manipulés par eux. Cela découlerait de la manière par laquelle ils travailleront avec des ordinateurs pour explorer de tels concepts. C'est cette exploration active, coopérative (élève à élève et élève à enseignant) et non imposée qui constitue la « philosophie LOGO ».

Papert réussit à convaincre, car il possède de très bons arguments. Mais la théorie pose de nombreux problèmes. Trop peu de preuves expérimentales viennent encore appuyer cette théorie ; les théories de Piaget sur l'éducation sont utilisées dans un contexte éducatif que Piaget lui-même n'avait jamais envisagé ; et il reste des domaines de recherche de solution à des problèmes (même en mathématiques !) que LOGO ne peut traiter.

Comme les enseignants utilisent de plus en plus LOGO dans les salles de cours, ils découvrent que tout ne fonctionne pas de la façon décrite par Papert, et ils n'obtiennent pas les résultats qu'ils espéraient. Un danger de désillusion existe donc. Mais, après avoir fait la part des choses en ce qui concerne les déclarations trop enthousiastes sur les possibilités de ce langage, LOGO demeure une excellente manière d'introduire des notions informatiques, d'explorer certains types de concepts et d'apprendre à résoudre divers problèmes.

Versions autorisées
Voici trois versions du LOGO, pour le Commodore 64 (Terrapin-MIT), pour le Sinclair Spectrum et pour les ordinateurs Atari (LCSI). Bien que coûteuses, il s'agit des versions de LOGO autorisées par les fabricants, et qui ressemblent de près à la version initiale mise au point par le MIT. Mais il existe des programmes LOGO moins chers sur le marché.

LOGO, sur les micros actuels, n'a pas assez de mémoire de travail et est exécuté trop lentement. On pourrait même dire que c'est un langage qui attend que le matériel soit à son niveau. Mais en tant que langage d'apprentissage, il n'a pas de rival sérieux.

A qui s'adresse logo?

Qui peut tirer profit de l'apprentissage de la programmation en LOGO ? Nous avons le sentiment que de nombreuses personnes, même des programmeurs expérimentés, peuvent apprendre avec cette programmation, dont :

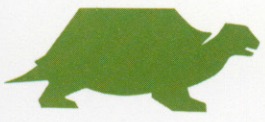
— toute personne sans expérience des ordinateurs ou sans expérience de programmation ;
— toute personne qui aime jouer avec les ordinateurs, et qui pense que les ordinateurs sont amusants à utiliser ;
— toute personne frustrée par le manque de puissance d'expression dans les autres langages informatiques ;
— toute personne s'intéressant de près ou de loin à l'enseignement ;
— toute personne qui désire découvrir des aspects plus évolués de l'informatique, en particulier l'intelligence artificielle.

Cela dit, nous devons rappeler que, comme le BASIC et le code machine, LOGO ne s'adresse pas à tout le monde. Il n'est pas particulièrement indiqué pour :

— toute personne qui associe exclusivement l'ordinateur au « travail ». Certains langages sont conçus pour travailler, comme des chevaux de labour. Mais vous ne choisirez certainement pas un cheval de labour pour faire une balade à la campagne ;
— toute personne qui a besoin d'une grande vitesse d'exécution. LOGO utilise beaucoup de mémoire, et il est très lent sur la génération actuelle d'ordinateurs. (Sur des programmes comparables, LOGO est exécuté deux fois plus lentement que BASIC.)

Cependant, même pour ces groupes de personnes, la connaissance de LOGO peut être rentable. Il peut servir à introduire une solution et à préparer sa traduction dans un autre langage.

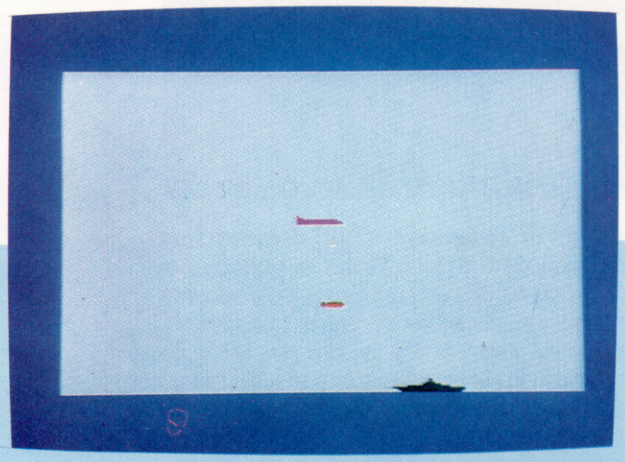
J'arrive...



la Tortue

Exocet sur le Vic 20

Pour tous ceux qui ont encore les Malouines entre les manettes, c'est l'occasion de prendre connaissance d'un jeu écrit en BASIC par Pierre Monsaut, pour le micro-ordinateur Vic 20.



Un porte-avions ennemi s'est aventuré dans les eaux territoriales et refuse d'obéir aux sommations. Aux commandes de votre Mirage 2000, vous devez absolument le détruire avant qu'il ne menace votre base. Tapez une touche quelconque pour tirer.

```
10 REM **********
20 REM * EXOCET *
30 REM **********
35 GOSUB 2000
40 GOSUB 1000
100 PRINT TAB(A)A$
105 IF B>17 THEN PRINT TAB(18)N2$;CHR$(1
9);:BB=0:GOTO 120
110 PRINT TAB(B)B$;
120 A=A-1
130 IF A<0 THEN PRINT D$+N2$+CHR$(19);:A
=19
140 BB=BB+0.2
150 B=INT(BB)
160 GET X$
170 IF X$<>"" AND EX=0 THEN EX=A+7923:NX
=NX-1
180 IF EX<>0 THEN  300
190 FOR I=1 TO 10
200 NEXT I
210 GOTO 100
300 EX=EX+21
310 IF EX>8184 THEN 400
320 C=PEEK(EX)
330 IF C<>32 THEN GOSUB 700
340 POKE EX-21,CN
350 POKE EX,CX
360 POKE EX+M,XC
370 GOTO 100
400 POKE EX-21,CN
410 EX=0
420 IF NX=0 THEN 500
430 GOTO 100
500 PRINT CHR$(147)
505 IF S>R THEN R=S
510 GET X$
515 IF X$<>"" THEN 510
520 PRINT:PRINT:PRINT
525 PRINT CHR$(31)
530 POKE 36869,240
535 PRINT TAB(5)"SCORE :"S
540 PRINT:PRINT:PRINT
550 PRINT TAB(5)"RECORD :"R
560 PRINT:PRINT:PRINT
570 PRINT TAB(5)"UNE AUTRE ?"
580 FOR I=1 TO 300:GET X$:NEXT I
600 GET X$
610 IF X$="" THEN 600
620 IF X$<>"N" THEN :POKE 36869,254:GOTO
 40
630 PRINT CHR$(147);
640 POKE 36879,27
650 END
700 POKE EX-21,CN
705 S=S+10
710 POKE EX,6
715 POKE EX+M,2
720 FOR I=1 TO 30
725 X=INT(RND(TI)*4)
730 Y=INT(RND(TI)*6)
740 XY=7680+(22-Y)*22+B+X
750 POKE XY,6
760 POKE XY+M,2
770 NEXT
780 FOR I=1 TO 200:NEXT
800 NX=NX+1:PRINT CHR$(147);:GOTO 100
1000 PRINT CHR$(147);
1010 M=30720
1020 B$=CHR$(144)+CHR$(32)+CHR$(64)+CHR$
(65)+CHR$(66)+CHR$(19)
1030 A=19:S=0:BB=0:B=0:D$=""
1060 FOR I=1 TO 10
1070 D$=D$+CHR$(17)
1080 NEXT
1085 D1$=D$+CHR$(17)
1090 A$=CHR$(156)+D$+CHR$(67)+CHR$(68)+C
HR$(32)+CHR$(19)+D1$+D$
1100 N2$=CHR$(32)+CHR$(32)+CHR$(32)
1120 EX=0:CX=5:XC=2:NX=20:CN=32:RETURN
2000 PRINT CHR$(147);
2010 POKE 36869,254
2020 POKE 52,24:POKE 56,24
2040 FOR I=0 TO 55
2050 READ A
2060 POKE 6144+I,A
2070 NEXT
2080 FOR I=0 TO 7
2090 POKE 6400+I,0
2100 NEXT
2110 POKE 36879,30:RETURN
3000 DATA 0,0,0,0,7,255,255,127
3010 DATA 16,16,56,252,255,255,255,255
3020 DATA 0,0,0,0,224,255,252,248
3030 DATA 0,0,0,0,0,63,127,255
3040 DATA 0,0,0,1,3,255,255,255
3050 DATA 0,0,0,0,125,255,125,0
3060 DATA 8,33,128,10,0,40,0,16
```

Nouveautés musicales

L'électronique fait évoluer la musique. Aujourd'hui, le codage numérique du son — dit échantillonnage — permet aux musiciens d'utiliser n'importe quel son pour produire de la musique.

Rappelons les caractéristiques d'un oscillateur simple. Lorsqu'un objet physique — qu'il s'agisse de l'aile d'une abeille ou des cordes vocales humaines — vibre, l'air avoisinant se dilate et se contracte très rapidement; un signal est alors perçu comme un son par l'oreille et le cerveau humains. Si une tension électrique est appliquée au moyen d'un modulateur (semblable à une bobine d'induction d'une automobile) à une fine lamelle métallique, le métal vibre, ce qui crée le signal le plus simple, l'onde sinusoïdale. La note, ou la fréquence de la vibration, dépend de la tension appliquée et, à un certain degré, de la densité de la lamelle métallique. Ce minuscule générateur sonore se nomme un oscillateur. Le contrôle de la tension a été la principale méthode utilisée pour produire de la musique synthétisée depuis des décennies.

L'interface MIDI, initialement introduite en 1983, est une unité qui est conçue de façon à permettre à un système numérique (comme un ordinateur) d'en commander un autre — un synthétiseur, par exemple. Son développement est issu des découvertes faites en musique électronique au cours de la dernière décennie.

Pendant des années, les studios d'enregistrement renfermaient plusieurs pièces différentes d'équipement de traitement sonore — un ensemble impressionnant de filtres et d'unités de réverbération était souvent perçu comme la preuve de la qualité du studio. De même, sur scène, un joueur de synthétiseur des années soixante-dix devait être entièrement entouré d'une série de claviers comportant une multitude de boutons.

Pour se représenter ce qui se produit dans un studio d'enregistrement bien équipé, nous n'avons qu'à penser au jeu du téléphone arabe. Dans ce jeu, une phrase est transmise d'une personne à l'autre. Le dernier joueur récite la phrase qu'il a entendue. Une phrase très simple peut ainsi avoir été transformée en une série de mots insensés, ou vice versa. Un processus similaire survient dans un studio d'enregistrement, mais ici, la phrase initiale est un ensemble de sons musicaux.

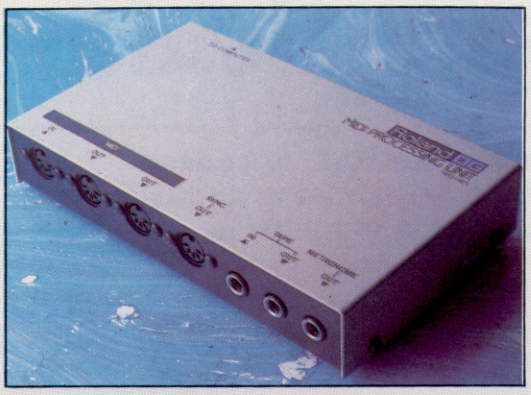
La chaîne de joueurs, dont chaque membre produit une version de la phrase initiale, est remplacée par un groupe d'unités de traitement sonore, chacune pouvant être commandée pour effectuer une tâche spécifique. L'emploi de cet équipement impliquerait normalement la présence d'un panneau central de connexions, ou la connexion directe des unités. Les commandes seraient calibrées manuellement en quelques secondes par un ingénieur du son expérimenté; mais les problèmes de synchronisation et de communication inhérents à ce type de connexions peuvent être facilement imaginés.

Jumelles
Alannah Currey, Tom Bailey et Joe Leeway, des Thompson Twins (d'avant en arrière). Initialement ils formaient un groupe de sept musiciens. Ils se produisent maintenant en trio avec des motifs rythmiques enregistrés.

Le joueur de clavier des années soixante-dix avait un autre problème. La difficulté immédiate n'était pas de produire successivement un son de chaque instrument — le joueur n'avait qu'à déplacer une main d'un clavier à l'autre. Mais les problèmes se posaient plus fréquemment lorsque deux claviers devaient être joués simultanément; les organistes d'église savent comment le faire depuis des siècles.

Mais le développement le plus important des années soixante-dix se situa au niveau des unités de génération sonore, à l'intérieur du synthétiseur, et au niveau des techniques instrumentales en général. L'émergence du synthétiseur de basse commandé par le clavier est un bon exemple de ce phénomène.

Au cours de ces années, les joueurs de basse funk avaient développé un niveau de virtuosité qui rivalisait avec celui des guitares solos; à la fin de la décennie, le synthétiseur de basse



Centre de contrôle
L'interface MIDI Roland MP401 est une unité évoluée qui connecte des micro-ordinateurs à des synthétiseurs numériques. Il contrôle toutes les fonctions de synchronisation interne et externe, l'entrée/sortie de bande, et la sortie du synthétiseur. Cela signifie que l'ordinateur central est uniquement responsable de l'envoi et de la réception de commandes et de la gestion de la mémoire. Le MP401 fonctionne avec l'IBM PC et l'Apple II et IIe. On attend une version Commodore.

commandé par le clavier fit son apparition. Cela entraîna d'autres problèmes pour le joueur de clavier, qui devait alors remplir les fonctions de joueur de basse — travail de synchronisation de la section rythmique — et ne jouait plus de musique strictement « clavier ». De quoi frustrer un vrai musicien. De nouveaux synthétiseurs analogiques firent leur apparition sur le marché, et des sons de trompette, de saxophone et de batterie devinrent disponibles. De plus en plus de joueurs de clavier ont opté pour un dispositif simple pour faire face à ces nouvelles responsabilités. Il s'agissait du séquenceur.

Un séquenceur est un dispositif qui assemble plusieurs phrases musicales en un motif préétabli. Son fonctionnement implique l'utilisation de tensions contrôlées qui sont transmises par un oscillateur produisant une série de sons de différentes fréquences. La vitesse de vibration de la lamelle métallique est fonction de la tension, et la forme d'onde résultante produit un son très aigu. Un séquenceur est utilisé pour commander l'oscillateur. Sa fonction est nécessaire, car la musique est rarement composée de séquences sonores ininterrompues — de courts intervalles ou de longs silences sont nécessaires, pour créer des motifs rythmiques ou pour former la structure globale de la musique. Un intervalle de motif est produit lorsque l'unité de commande envoie une tension zéro à l'oscillateur. Le séquenceur a pour fonction de garantir que cet intervalle surviendra exactement au même endroit à chaque répétition du motif.

A la fin des années soixante-dix, peu de synthétiseurs disposaient des fonctions complètes d'un séquenceur, mais les musiciens s'emparèrent rapidement de ce que l'industrie pouvait leur offrir. La musique disco produite par Giorgio Moroder avec Donna Summer est un exemple de ce type de musique; les groupes utilisant des synthétiseurs développèrent un style entièrement nouveau pour ce nouvel instrument. Il n'était plus nécessaire de jouer chaque note manuellement. A la place, une séquence entière pouvait être appelée et interrompue à l'aide d'un bouton. Pendant ce temps, le joueur pouvait se déplacer autour du clavier, danser au rythme du séquenceur et revenir à un autre clavier pour jouer une mélodie ou un groupe d'accords. Aujourd'hui, le style général de certains groupes est influencé par l'existence du séquenceur.

Steve Cross

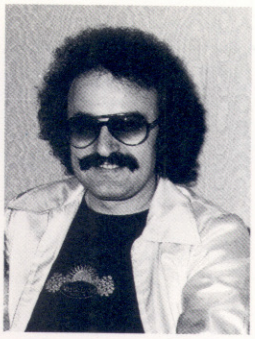
Branche-toi et joue!

Des sons électroniques peuvent être produits en programmant la sortie de générateurs de notes et/ou de bruit (c'est ainsi que l'on crée un son sur un micro-ordinateur); ou, à partir d'un échantillonnage de sons réels, en créant un modèle numérique de sons et en utilisant ce modèle pour recréer le signal des sons à l'aide d'un convertisseur numérique/analogique alimentant un haut-parleur. L'échantillonnage permet d'obtenir une image sonore précise (selon le nombre d'échantillons de signaux stockés), qui serait difficile à programmer autrement, sauf par essais et erreurs — essayez de produire un rythme de batterie sur votre micro!

Lors de l'apparition des synthétiseurs numériques, la conception de ces outils fut souvent modelée sur leurs prédécesseurs analogiques. Les musiciens découvrirent que leurs connaissances étaient encore mieux exploitées par ces nouveaux instruments, et c'est dans ce domaine qu'on se concentra surtout. Cela est démontré par la récente popularité de la batterie Linn, l'une des premières unités à utiliser des échantillons sonores — il s'agissait ici du jeu du batteur américain Steve Gadd. Sur la machine Linn, les motifs rythmiques sont enregistrés sous forme numérique, de manière similaire à celle des données informatiques sur disquettes. Les séquences résultant des 1 et des 0 sont alors codées sur des puces ROM. En utilisant une puce particulière, un musicien ou un régisseur pouvait reproduire le son initial comme s'il était joué en direct. Le grand avantage de la numérisation du son est que la sortie peut être modifiée à la console, changeant ainsi le motif, la vitesse, le volume, etc.

Arrivé à ce stade, nos deux situations initiales (le studio d'enregistrement et le joueur de synthétiseur) ont beaucoup de points en commun. Les travaux de studio reflètent l'engouement qui s'est manifesté ces dernières années pour la vidéo. La vidéo a introduit un nouveau style et a fait prendre conscience de l'importance de l'image : les producteurs exigent que la musique d'accompagnement reflète cette réalité.

Si la musique accompagnant une vidéo est produite par des instruments conventionnels, la synchronisation avec l'image est similaire aux techniques utilisées pour les films. Si, cependant, la musique est composée de sons et de séquences individuels produits par des synthétiseurs, le risque d'introduire des problèmes est considérable. Imaginons qu'une vidéo où un pot de fleurs tombe sur le sol et se brise. Le musicien travaillant sur la partition a produit une séquence qui s'accélère jusqu'au moment où le vase tombe, et il provoque un accord correspondant au moment de l'impact sur le sol. L'enregistrement démarre et la séquence commence, mais il devient rapidement évident que le rythme d'accélération a été mal calculé : la séquence se termine alors que le vase se trouve toujours sur la table. Le musicien essaie alors l'accord d'impact, qui est enregistré sur une piste différente. On reprend l'enregistrement, mais cette fois l'accord a un retard d'une fraction de seconde. Les musiciens ont besoin, d'une certaine manière, de relier les instruments numériques, de telle sorte que tout arrive au même moment. En réalité, on a besoin d'une interface numérique.

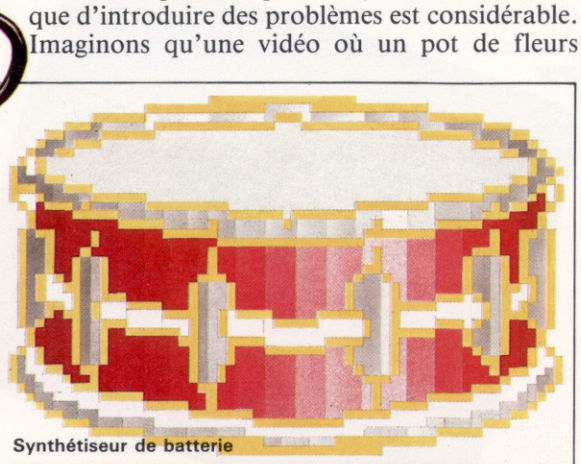


Duo dynamique
Donna Summer, avec le producteur Giorgio Moroder, firent partie des premiers artistes à utiliser des rythmes produits électroniquement et des synthétiseurs en musique enregistrée.
(Cl. Record Mirror.)

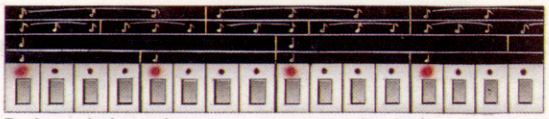
Le joueur de synthétiseur a des problèmes similaires sur scène. Son équipement comprend deux synthétiseurs numériques, construits par des fabricants différents, ainsi qu'une unité de batterie Linn. Il a des séquences préparées sur l'un des synthétiseurs et sur le Linn ; mais, puisqu'elles ne sont pas parfaitement synchronisées, il décide généralement d'utiliser le Linn automatiquement et de jouer l'autre manuellement. Le résultat est que son second synthétiseur, acheté pour la qualité de ses sons prédéfinis, n'est pas utilisé. Ce musicien a besoin de trouver un moyen pour relier ses instruments, afin de synchroniser l'ensemble.

Il faudrait aussi que le séquenceur de son premier synthétiseur joue les sons prédéfinis sur le second. De plus, l'équipement utilisé devrait être compatible avec des unités différentes de ses propres synthétiseurs. Il se peut qu'il ait à affronter un problème de synchronisation comme celui mentionné précédemment.

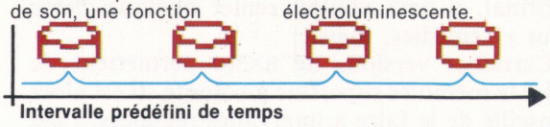
Dans le prochain article de cette série, nous examinerons en détail l'interface MIDI et nous étudierons d'autres exemples de techniques de séquences.



Synthétiseur de batterie



Rythme de batterie électronique
Une phrase de batterie peut maintenant être construite sur les touches du synthétiseur. Lors d'une production de son, une fonction parcourt le clavier par intervalles prédéfinis, jouant un rythme de batterie numérisé si cette touche a été pressée et illuminant sa diode électroluminescente.



Le facteur humain

Un aspect important de la conception des programmes est l'interface homme/ordinateur. Nous abordons ici l'étude des facteurs à prendre en compte lors de la conception de cette interface.

La programmation fut pendant longtemps un sujet hermétique réservé à un petit nombre d'initiés. L'utilisateur contemporain est, par comparaison, un petit gâté. Les fabricants ne lui demandent plus d'ingurgiter le maniement du code machine. L'expression « l'ordinateur pour tous » signifie bien que le micro peut être programmé par un néophyte. En 1982, le « Alvey Committee », dans un rapport intitulé « Programme pour une technologie avancée de l'information », considérait l'interface homme/machine comme l'une des quatre priorités pour la recherche et le développement. La conception de logiciels, l'intégration à très grande échelle et les systèmes experts formant les trois autres.

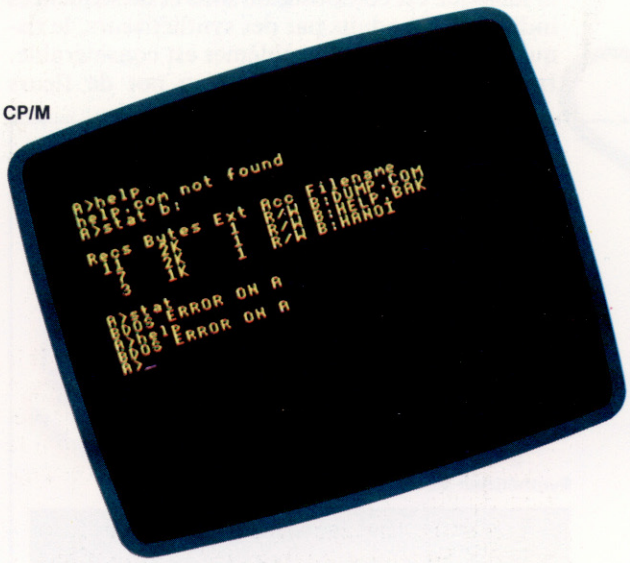
Dans toute application, l'interaction entre l'ordinateur et l'utilisateur, au cours de laquelle des données et des instructions circulent de l'un à l'autre, est de première importance. Ce dialogue a lieu par l'intermédiaire des périphériques d'entrées/sorties (E/S). Le clavier est la principale des entrées et l'affichage sur l'écran, celle des sorties. Les manettes de jeux, les « souris », les écrans tactiles et autres périphériques sont également des entrées. De son côté, l'ordinateur peut utiliser une imprimante, un générateur de sons ou de paroles, ou même un robot, pour exprimer les sorties.

Outre les contraintes imposées par les périphériques d'E/S utilisés, le dialogue entre l'homme et la machine est conditionné par le logiciel. Par exemple, le système d'exploitation de l'ordinateur contrôle de nombreux aspects de la gestion du clavier et de l'affichage. La cadence de répétition d'une touche, lorsqu'on la maintient appuyée, et l'intervalle entre les répétitions sont déterminés par le système d'exploitation. Ce dernier se charge également du tampon mémoire gérant le clavier pour garder trace des caractères frappés trop vite par rapport à l'affichage. C'est très important, car il conditionne la vitesse possible de saisie de l'information. La taille du tampon-clavier est essentielle, et elle devrait être connue de l'utilisateur. Le système d'exploitation CP/M, par exemple, ne peut mettre dans son tampon mémoire-clavier qu'un seul caractère; alors que de nombreux micros personnels permettent dix frappes ou davantage.

Mais les tampons mémoire-clavier causent également des problèmes. Un utilisateur habitué à un logiciel peut connaître les choix à faire dans une suite de menus. Il saura, par exemple, qu'il devra retenir les options 2 (menu principal), 5 (1er sous-menu), 3 (2e sous-menu), puis 4, puis 6. Étant sûr de ses choix, il frappera les réponses très vite, avant même que les menus successifs ne s'affichent. Avec un tampon-clavier à 10 caractères, l'utilisateur obtiendra l'effet voulu. Avec un tampon à un seul caractère, le temps pour afficher le deuxième menu sera probablement trop long par rapport au temps de toute la séquence des choix frappés au clavier. Ainsi, au lieu de retenir l'option n° 5, l'affichage correspondra à la dernière frappe, le 6, le tampon n'ayant pas retenu le choix correspondant.

Un tampon mémoire-clavier trop grand pose également d'autres problèmes. Un menu qui tarderait à se manifester à l'écran (cela peut arriver par exemple lorsque les choix conduisent à un fichier en cours de lecture) peut faire penser qu'il ne se passe rien. L'utilisateur nerveux peut être alors tenté de taper à nouveau l'ordre donné et, ensuite, sur toutes les touches, dans l'espoir de débloquer la situation. Il en résultera autant de caractères aberrants enregistrés dans le tampon, que le programme essaiera en vain d'exécuter!

CP/M



La récupération d'espace mémoire, qui consiste à vider les registres mémoire pour libérer de l'espace de travail, constitue une autre source de problèmes. Cette opération peut provoquer un arrêt apparent dans le déroulement du programme. Si l'utilisateur pense que cela est anormal, il peut vouloir tenter quelque chose pour y remédier, mais...

Certaines versions de BASIC permettent de vider la mémoire superflue à volonté. Il est alors conseillé de le faire à intervalles réguliers. Cela

concerne le programmeur, l'utilisateur recevant alors un message d'attente.

La manière dont un langage de programmation traite des E/S influence l'interface homme/ordinateur. Les ressources uniques du BASIC pour le maniement des chaînes de caractères permettent une utilisation sophistiquée de ces dernières dans le dialogue homme/machine, surpassant en cela des langages comme le PASCAL. Les versions du BASIC qui comportent des commandes pour l'adressage du curseur favoriseront une meilleure présentation de l'écran. De même les BASIC avec commandes graphiques. Le BASIC est bien loti pour ce qui concerne les commandes d'E/S — INPUT et PRINT font tout à fait l'affaire pour des programmes simples. Mais pour un contrôle effectif des entrées (comme pour créer un masque de saisie avec des zones protégées), essayez plutôt GET$, INKEY$, INPUT$ () ou des commandes similaires. PRINT USING est une commande extrêmement souple pour formater l'état des sorties. Elle est de grande valeur pour aligner les décimales d'un nombre et pour justifier des colonnes d'un texte.

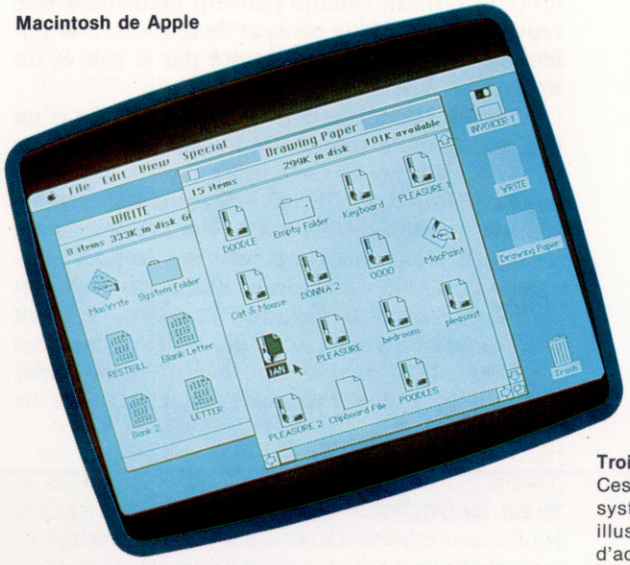
Micronet 800
« menu annuaire électronique »

L'utilisateur est l'élément le plus imprévisible des systèmes homme/machine. Cependant, à l'image de toute organisation, l'utilisateur présente des performances caractéristiques, dont la prise en compte est essentielle avant la conception de l'interface homme/ordinateur.

L'homme partage avec l'ordinateur la propriété d'être un « analyseur d'informations ». Cela dit, l'être humain est limité quant à la quantité d'informations qu'il peut enregistrer dans sa « mémoire de travail ». Il est ainsi admis que sept éléments d'information différents peuvent être assimilés à la fois par l'homme. Si l'information à emmagasiner consiste en des caractères sans rapport entre eux, chaque élément mémorisable ne pourra comprendre qu'un seul de ces caractères. Par contre, si les caractères sont groupés sous forme de mot ou de structure intelligible, chaque structure aussi complexe soit-elle pourra être retenue en tant qu'élément d'information individuelle.

Il existe diverses méthodes pour aider l'utilisateur à structurer les informations destinées à l'ordinateur. L'une consiste à les apparenter à des méthodes de travail familières; c'est la représentation des tâches de bureau avec Lisa, par exemple. De la même manière, un tableur d'applications financières sera souvent organisé pour rappeler un livre, avec des pages et un index. Une autre méthode sera l'apprentissage de nouvelles structures. En montrant de manière répétitive divers exemples, et en exposant à fond la méthode, le programme lui-même servira à apprendre à l'utilisateur à structurer l'information. L'inconvénient de ce genre d'approche est le coût en temps et en efforts. Des instructions

Macintosh de Apple



détaillées, des pages-écrans d'assistance et des repères dans le déroulement du programme peuvent fournir une sorte de formation au cours de l'utilisation même du logiciel.

En dernier lieu, l'information peut être présentée sous la forme de schémas organisés. Une présentation appropriée avec éventuellement des couleurs peut aider l'œil humain à repérer sur l'écran l'information recherchée. Il suffit, pour s'en convaincre, de penser au Vidéotext et à ses codes graphiques si particuliers. Sur une page donnée, selon un tel système, les lignes d'en-tête et de bas de page seront dans la même couleur; la couleur du fond sera unie; le texte étant affiché selon deux autres couleurs, une par paragraphe. Les mots clés pourront apparaître en double brillance par exemple. Le but est alors de permettre un repérage immédiat de l'information utile en ignorant les autres parties de la page ne comportant pas d'intérêt immédiat. Coder de manière distinctive par des couleurs différentes peut aussi tout embrouiller, si l'on en abuse. Des tests ont ainsi montré que l'utilisateur perdait beaucoup de temps à lire et à relire des paragraphes entiers pour essayer de comprendre l'intérêt d'un changement de couleur d'affichage arbitraire! Une bonne ligne de conduite est de ne jamais utiliser plus de quatre couleurs à la fois.

Trois degrés
Ces photographies de système d'exploitation illustrent trois degrés d'accessibilité de l'ordinateur. Dans la première, un utilisateur essaye de communiquer avec le système d'exploitation CP/M. Ce dernier ne comporte pas de caractéristiques d'assistance, et suppose donc une connaissance approfondie de ses commandes avant utilisation. La deuxième photo illustre un système se déroulant par menus. Les options sont numérotées de manière précise, et l'utilisateur fait son choix en tapant un numéro. Les menus ne sont pas explicites et l'utilisateur doit en connaître les fonctions avant de les lancer. La troisième photo montre le système d'exploitation Macintosh. Ce dernier propose des repères visuels et des affichages graphiques, ainsi que des menus simples et détaillés. (Cl. Ian McKinnell.)

Câblage

Nous allons examiner l'accès au port utilisateur de façon à vous permettre de contrôler des phénomènes physiques comme la chaleur, la lumière et la force, et de commander des dispositifs externes.

De nombreux micros domestiques populaires possèdent des ports utilisateur qui permettent d'accéder à la topographie mémoire au moyen d'une série de connexions électriques. Tout système numérique repose sur le fait que les 1 et les 0 du système binaire peuvent facilement être représentés par deux niveaux de tension. Normalement, un zéro est représenté par 0 volt et un un par + 5 volts.

Chaque adresse mémoire est composée d'un groupe de huit cellules individuelles, chacune ayant un niveau de tension de 0 ou 5 volts. La configuration de ces niveaux de tension détermine alors le nombre qui est stocké dans cette adresse mémoire. Les connexions externes du port utilisateur sont reliées électriquement à une ou à plusieurs adresses de la mémoire du micro ; en lisant des valeurs ou en écrivant des valeurs dans ces adresses, nous pouvons contrôler ou commander des systèmes électriques situés à l'extérieur de l'ordinateur.

Il existe deux types de connexions de port utilisateur. Certains ports ont des groupes séparés de broches (huit pour l'entrée et huit pour la sortie) reliés à deux adresses en mémoire. D'autres utilisent les mêmes broches pour gérer à la fois l'entrée et la sortie. Dans ce premier article, nous examinerons le deuxième type de configuration, utilisé par le Commodore 64.

Les registres de direction de données

En plus de posséder une adresse reliée aux huit broches du port utilisateur, les micros munis de ports bi-directionnels utilisent une seconde adresse mémoire, dite registre de direction de données (DDR). Ce registre détermine si les huit lignes envoient ou reçoivent des données. Un état un dans le DDR met une ligne dans le mode sortie, et un zéro permet de recevoir une entrée. Pour

Bien branché

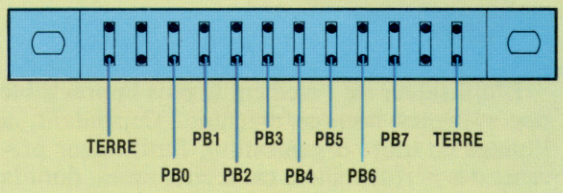
Nous avons commencé notre projet en faisant des connexions pour le Commodore 64. Celles-ci seront utilisées pour relier les machines au monde extérieur via un format commun de huit lignes de données et d'une ligne de mise à la terre de chaque côté. Vous aurez besoin pour le Commodore 64 de :

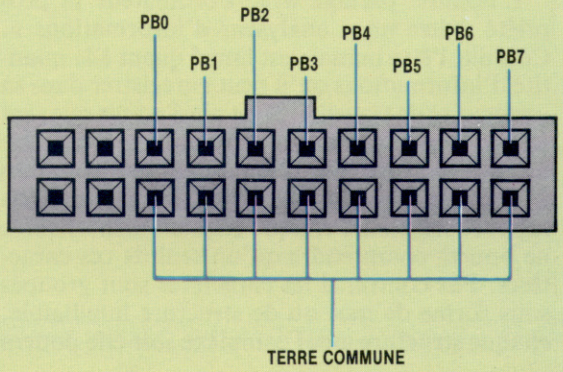
— un connecteur plat à 24 broches de 0,15 pouce ;
— un câble-ruban à 10 broches (environ 1 m) ;
— un fer à souder et de la soudure.

Les diagrammes des broches de port utilisateur des machines montrent l'emplacement des 10 lignes (8 pour les données, 2 pour la mise à la terre) dont nous avons besoin. A titre d'exemple, autre que le Commodore 64, sachez pour votre information que, par exemple, le connecteur IDC du BBC Micro possède une languette de localisation sur l'un des côtés et qu'elle se divise en deux parties inégales : tenez le connecteur verticalement, en plaçant la languette du côté opposé par rapport à vous et sa section de fixation dirigée vers le haut, et enfilez-y environ 2 cm de câble-ruban en plaçant la bande rouge à votre droite. Fermez le connecteur sur le câble en exerçant une pression assez forte (éventuellement avec un étau ou des pinces). Séparez et écartez 10 lignes à l'autre extrémité comme sur l'illustration, puis dénudez les extrémités des autres lignes.

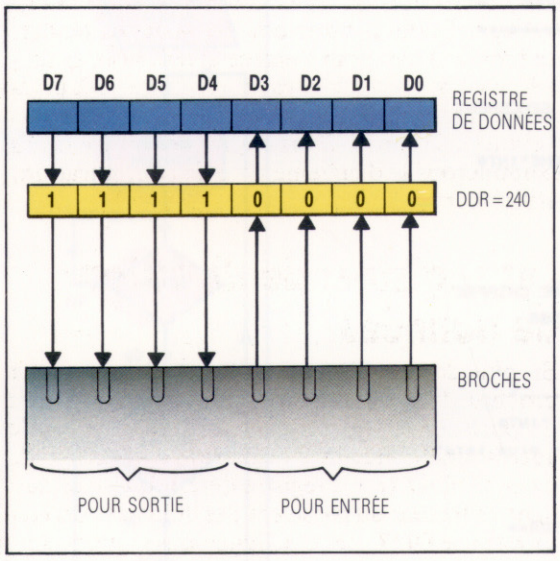
Revenons au Commodore 64 : identifiez l'un des côtés du connecteur comme étant le haut (et placez toujours ce côté vers le haut lors d'une connexion à la machine). Dénudez les deux extrémités des 10 lignes, et soudez le câble aux broches inférieures du connecteur plat en suivant le diagramme de brochage.

Utilisez les programmes tests et un multimètre pour vérifier vos conducteurs.





mettre les huit lignes du port utilisateur en sortie, le DDR doit être mis à 255 (c'est-à-dire 11111111 en binaire). De façon similaire, les huit lignes peuvent être réservées pour l'entrée, en réglant la valeur du DDR à zéro. Les huit lignes peuvent être configurées dans toute combinaison de lignes d'entrée ou de sortie, en définissant la valeur appropriée du DDR. Les quatre lignes les plus significatives du port utilisateur pourraient être mises en mode sortie, et les moins significatives, en mode entrée, en plaçant la valeur 240 (11110000 en binaire) dans le DDR.



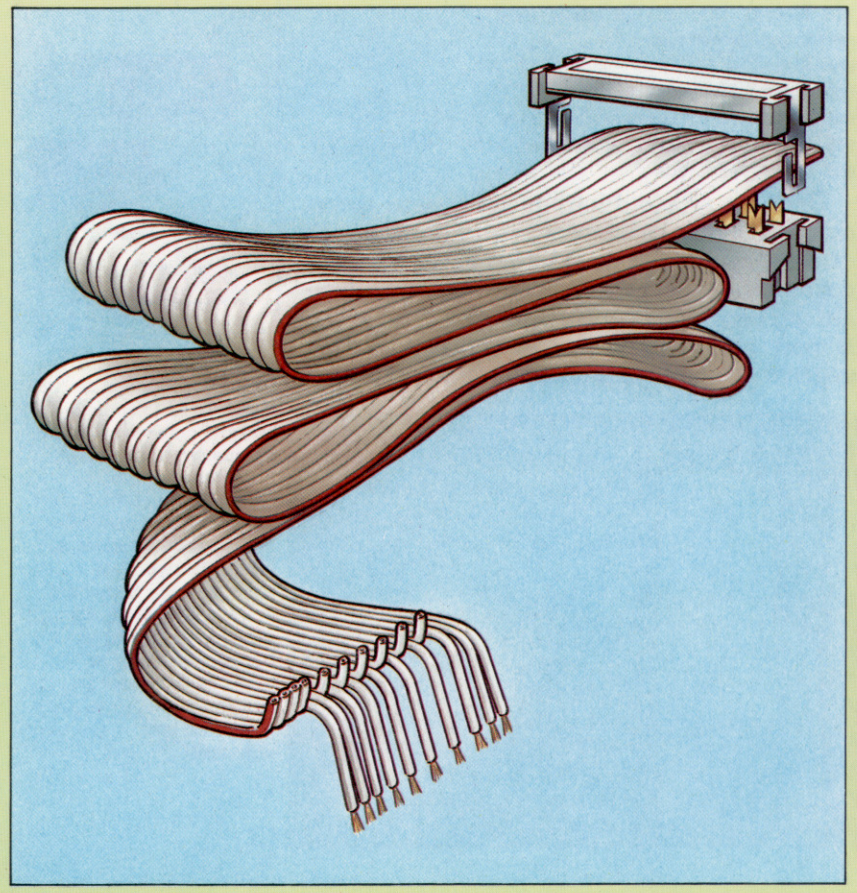
Kevin Jones

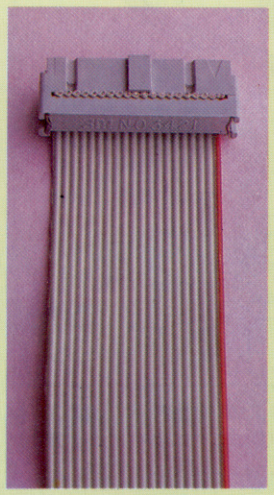
Les registres de données et de direction de données ont les adresses suivantes :

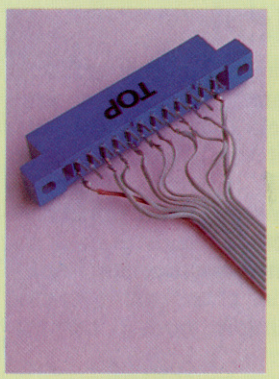
Type de micro	Registre de données	Registre direction données
Commodore 64	$DD01 (56577 décimal)	$DD03 (56579 décimal)

Le programme suivant règle le port utilisateur de façon à pouvoir utiliser les huit lignes pour l'entrée, et affiche le contenu du registre de données :

```
4 REM**************C64***************
5 REM*         AFFICHAGE DATREG      *
6 REM**************C64***************
10 DIM A$(10) :A$(0)="0":A$(1)="H"
20 DATREG=56577:DDR=56579
30 POKE DDR,0: REM = INPUT ONLY
50 :
100 PE=PEEK(DATREG):GOSUB 500
150 PRINT"DATREG"=" ;PE;"=" ;B$
200 GOTO 100
300 :
499 REM*****************************
500 REM*   CONVERSION BINAIRE   *
501 REM*****************************
550 .B$= "" :N=PE
600 FOR D=1 TO 8
650 N1=INT(N/2):R=N-2*N1
700 B$=A$(R)+B$:N=N1
750 NEXT D:RETURN
```





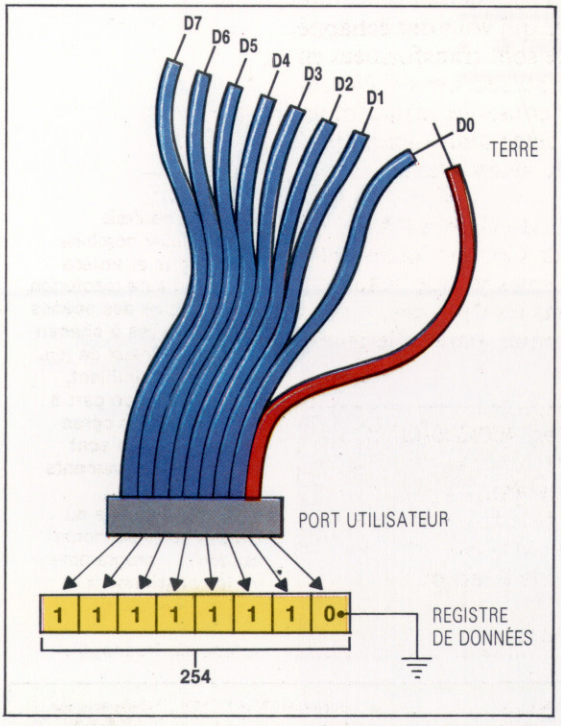


Lignes de connexion
Les conducteurs du port utilisateur du Commodore 64.

L'exécution du programme illustre le contenu normal du registre de données, 255 (illustré à l'écran par 11111111), ce qui signifie que les huit lignes sont à l'état 1, et que les huit cellules de DATREG sont au niveau de tension + 5. Si vous branchez maintenant un câble dans le port utilisateur, vous pouvez l'utiliser pour changer les niveaux de tension des lignes, et, par conséquent, pour changer le contenu numérique de DATREG.

Nous avons câblé dix lignes au port utilisateur : huit d'entre elles sont des lignes de données — une ligne pour chaque bit de DATREG —, et les autres fournissent une mise à la terre au système. Relier une ligne de données à une ligne de mise à la terre ramène le niveau de tension de la ligne de données à zéro, ce qui change les niveaux de tension de DATREG.

Si, par exemple, vous mettez D à la terre, pendant l'exécution du programme, vous verrez que DATREG passera à 254 (illustré sur l'écran par 11111110); ce qui signifie que la ligne la moins significative est au niveau de tension zéro (0 V), tandis que les autres sont à un niveau 1 (+ 5 V). Vous pouvez essayer différentes combinaisons de



lignes, afin de constater que vous pouvez, de cette façon, donner à DATREG toute valeur comprise entre zéro et 255.

Écriture de logiciel

La vérification des valeurs numériques (ou conditions) et le branchement résultant de cette vérification sont essentiels dans les processus décisionnels d'un ordinateur. Il est donc assez simple de concevoir un logiciel qui puisse contrôler une activité physique, via le port utilisateur. En testant la valeur du registre de données et en prenant les mesures appropriées, l'ordinateur peut répondre à des changements externes. A titre d'exemple simple, nous pourrions concevoir un programme qui vérifie qu'un numéro de téléphone a été composé correctement. Le numéro pourrait être composé en mettant à la terre certaines lignes de données connectées au port utilisateur; cela permettrait de produire la valeur numérique appropriée dans le registre de données pour chaque chiffre du numéro de téléphone. En raison des difficultés qu'implique la mise à la terre simultanée de plusieurs lignes de données, le programme attend que l'utilisateur appuie sur une touche avant d'analyser le contenu du registre de données.

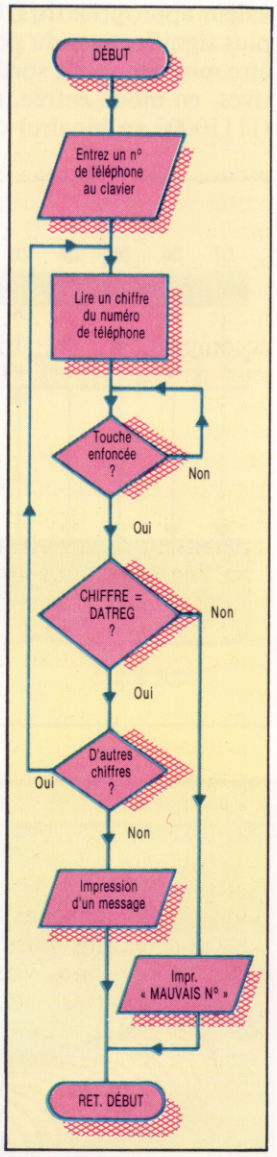
```
100 REM*************C64**************
101 REM*   NUMEROS DE TELEPHONE   *
102 REM*************C64**************
120 :
130 DATREG=56577:DDR=56579
140 POKE DDR,0:REM = ENTREE SEULEMENT
150 :
160 INPUT "NUMERO DE TELEPHONE";NT$
170 :
180 FOR K=1 TO LEN(NT$)
170 :
180 FOR K=1 TO LEN(NT$)
190 CH$=MID$(NT$,K,1)REM LIRE CHIFFRE
200 IF CH$<>" " THEN GOSUB 500
210 NEXT K
250 :
300 PRINT "-------DESOLE-------"
310 PRINT "Pas de réponse à ";NT$
320 PRINT "Veuillez rappeler plus tard"
350 PRINT:PRINT:RUN
400 :
500 REM**CONVERTIR ET VERIFIER**
550 CH=VAL(CH$)
600 PRINT"METTRE LE CHIFFRE SUR LES LIGNES"
650 PRINT"     ET TAPER UNE TOUCHE"
700 GET GT$:IF GT$=" " THEN 700
750 PE=PEEK (DATREG):IF PE=CH THEN
PRINT CH" OK":RETURN
800 :
850 PRINT"???MAUVAIS NUMERO???"
900 PRINT NT$"<>"LEFT$(NT$,K-1):PE
950 PRINT"??? ESSAYEZ DE NOUVEAU ???"
999 PRINT:PRINT:RUN
```

Dans cet article d'introduction, nous avons examiné l'entrée de données à partir d'une source externe, dans le but de modifier le déroulement des opérations des programmes. Dans le prochain article, nous nous pencherons sur la sortie par le port utilisateur et sur la conception d'un système de commande informatique simple.

Exercices de programmation

En utilisant le programme de numéro de téléphone comme exemple :

1. Écrivez un programme qui simule l'action d'une alarme contre le vol.
2. Écrivez un programme qui compte le nombre d'impulsions reçues par le port utilisateur pendant une période donnée.
3. Modifiez votre dernière réponse de façon à conserver un compte distinct pour chacune des lignes de données du port utilisateur.
4. Écrivez un programme qui simule l'action d'un cadenas à combinaison.
5. Écrivez un programme qui change la couleur de l'écran à partir du port utilisateur.



Lignes croisées
Le programme des numéros de téléphone accepte comme entrée un numéro (tout format, toute longueur), et compare ses chiffres un par un avec le contenu de l'adresse mémoire du port utilisateur. Les espaces dans le numéro entré sont ignorés. Le programme attend la frappe d'une touche avant de comparer un chiffre entré avec le chiffre du port utilisateur.

Magie noire

Dû à Bill Williams, Necromancer, de Synapse Software, est une véritable pièce de théâtre en trois actes bien structurés évoluant entre le « Bien » et le « Mal ».

OUVERTURE. Les deux versions (Atari et Commodore 64) se chargent en mémoire sans difficulté; la lenteur du lecteur de disquettes Commodore permettra d'apprécier pleinement les sons, très bizarres, créés par le programme, lesquels font un peu penser à un orchestre électronique en train de s'accorder! Le générique est accompagné d'une musique qui sait tirer le meilleur parti des possibilités sonores de chaque appareil.

ACTE I : *la Forêt*. Le drame commence à « l'âge des Ténèbres », où « Tetragorn, le sorcier maudit, règne en maître ». Vous jouez le rôle d'Illuminar, un druide présenté comme étant le « défenseur de la vérité et le protecteur de la race humaine ». Comme vous vous en seriez douté, ce n'est pas une tâche facile. Le druide fait son apparition au milieu d'un écran devenu noir. Des étoiles flottant autour de lui le protègent de centaines de petits ogres qui se ruent sur lui en agitant des coutelas géants, au son d'une musique de circonstance. Grâce à votre manche à balai, vous pourrez les détruire (et marquer des points) à l'aide d'une « nuée » magique, qui réduit à néant vos adversaires.

De surcroît, il suffit de la placer à l'endroit voulu et d'appuyer sur le bouton de mise à feu, pour qu'aussitôt des arbres se mettent à pousser : vous pourrez ainsi faire naître une forêt qui vous sera utile plus tard. Vous devrez bien sûr protéger ces plantations contre les ogres, ainsi que contre les araignées envoyées par Tétragorn. Gardez aussi un œil sur votre force : elle s'accroît quand vous tuez une araignée, et faiblit quand l'une d'elles vous pique. Au sixième niveau de jeu, le premier acte prend fin lorsque le druide est terrassé par une attaque générale des arachnidés. A ce moment, l'action s'interrompt : le programme compte le nombre d'arbres que vous avez eu le temps de planter, et passe ensuite à l'acte suivant.

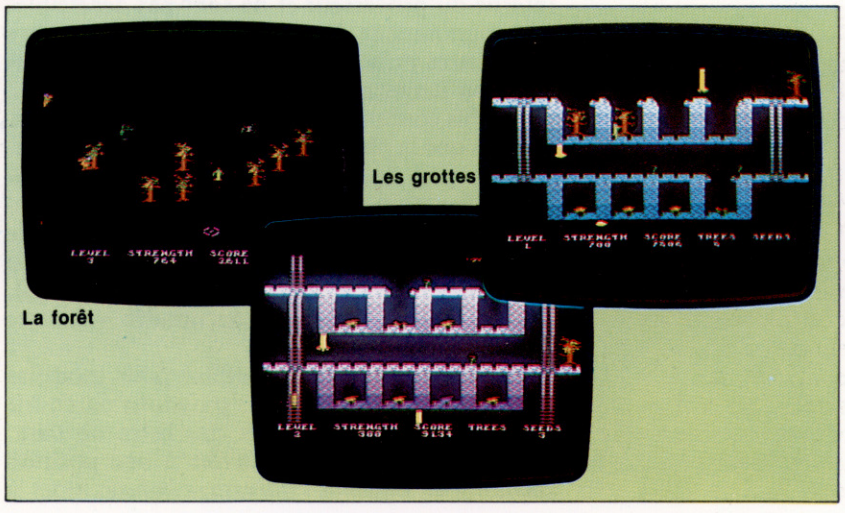
ACTE II : *Dans les souterrains*. C'est là que les araignées pondent. Il y a cinq niveaux différents, et chacun d'eux comporte huit grottes, réparties en deux niveaux de quatre. Les œufs qui vont éclore sont repérables à leurs changements de couleurs. C'est là que les arbres, que vous avez plantés au premier acte, vont vous être utiles : grâce à votre « nuée magique », vous les prenez dans le coffre où ils sont déposés, et vous les déplacez jusqu'au-dessus d'une grotte.

Il faut aussi se méfier des « mains du destin » : elles pendent au plafond et se saisissent de tout ce qui passe à leur portée : druide, arbres ou points d'interrogation. Ces derniers représentent des trésors mystérieux, et il faut en acquérir au moins un avant de pouvoir passer au niveau suivant, grâce à une échelle.

ACTE III : *le Repaire du Nécromant*. L'action atteint son paroxysme dans un cimetière aussi sombre qu'inquiétant. Les pierres tombales sont celles du Nécromant; il faut les détruire pour qu'il ne puisse plus se réincarner. Pour cela, le druide doit marcher sur elles, tandis que la « nuée » permet d'anéantir chaque nouvelle incarnation de l'ennemi. Mais un affrontement entre le « Bien » et le « Mal » ne saurait se réduire à cela. Toutes les araignées qui vous ont échappé au cours du deuxième acte sont transformées en « zombies », et viennent au secours du Nécromant. Il vous faudra donc conserver le maximum d'énergie pour le dernier acte; mais sachez qu'il est impossible d'y accéder avant d'avoir triomphé des précédents.

Des deux versions, celle destinée à l'Atari est de loin la meilleure. Celle du Commodore est plus lente, le son et les graphismes y sont beaucoup moins spectaculaires. Dans les deux cas, il vous faudra des manches à balai particulièrement rapides et précis.

Necromancer : pour Atari 400/800 (32 K) et pour Commodore 64.
Éditeurs : Synapse Software.
Auteurs : Bill Williams (Atari), Scott et Steve Coleman (Commodore 64).
Manches à balai : de type Atari ou Commodore.
Format : disquette et cassette.



Le plaisir de l'œil
Necromancer combine plans-objets et arrière-plans en haute résolution pour produire des scènes extraordinaires à chacun des trois niveaux de jeu. Les arbres scintillent, la nuée magique part à toute allure, les ogres et les araignées sont animés de mouvements très élaborés, le Nécromant crache du feu; autant d'éléments qui font du programme un jeu passionnant et une parfaite réussite graphique.
(Cliché Liz Heaney.)

Langage de processeur

Voici le premier d'une série d'articles traitant du langage d'assemblage du processeur 6809, utilisé sur le Dragon et le Tandy Color. Commençons par expliquer le rôle des registres.

Un microprocesseur peut être considéré comme étant composé de trois éléments principaux : les registres, qui font partie de la mémoire à l'intérieur du processeur; une ALU (unité arithmétique et logique), où peuvent être effectuées certaines opérations mathématiques simples sur les données stockées en mémoire; et une unité de contrôle qui veille à ce que tout s'effectue en bon ordre et en temps voulu.

Au niveau inférieur, le microprocesseur répond à des signaux de tension appliqués à des connexions externes pour modifier son état interne (le contenu de ses registres), ou pour émettre ou recevoir d'autres signaux. Nous pouvons programmer le processeur en appliquant une séquence de nombres qui le font agir comme des instructions. Le niveau inférieur de programmation implique donc que l'on raisonne en termes de nombres binaires (ou hexadécimaux). Cela nécessite une connaissance de l'effet de chaque nombre, ou code instruction, sur le processeur.

Un processeur 8 bits, tel que le 6809, peut émettre et recevoir des nombres binaires à huit bits, ce qui correspond aux nombres décimaux compris entre 0 et 255. Beaucoup de ces nombres sont utilisés pour renvoyer à des adresses mémoire, qui, sur la plupart de ces processeurs, sont données sous la forme de nombres à 16 bits (donc compris entre 0 et 65 535). Bien sûr, quand il traite ces nombres, le processeur ne peut les transférer que par paquets de 8 bits.

Les registres du 6809

Les registres d'un processeur peuvent prendre différentes formes. Certains sont réservés à l'usage interne du processeur et ne sont pas accessibles au programmeur. On estime à quatre le nombre de registres principaux du 6809 que le programmeur en langage machine utilisera beaucoup.

Le plus couramment utilisé est l'*accumulateur*. C'est là que la plupart des données en cours d'utilisation sont stockées et manipulées. Par exemple, la fonction usuelle accomplie par l'instruction d'assembleur ADD consiste à additionner le contenu d'un emplacement mémoire spécifié au contenu déjà stocké dans l'accumulateur. Ainsi, une nouvelle valeur sera « accumulée » dans ce registre.

Le *registre d'index* est utilisé pour modifier des adresses, afin que nous puissions parcourir facilement des tableaux et des listes de data. Lorsqu'une instruction renvoie à une position mémoire en utilisant l'*adressage indexé,* alors le contenu de ce registre est additionné à l'adresse donnée pour spécifier l'*adresse effective* de la donnée requise. Pour parcourir un tableau de données, nous n'avons qu'à nous référer à l'*adresse de base* (celle du premier article du tableau), puis à incrémenter le registre d'index. Puisque les valeurs stockées dans ce registre sont normalement des adresses, les registres d'index ont généralement 16 bits de longueur, plutôt que 8.

Le *pointeur de pile* est le registre qui indique l'emplacement du haut de la *pile,* ce qui est pratique pour stocker des données et les récupérer rapidement. La pile est utilisée lorsqu'il est nécessaire de sauvegarder le contenu interne du processeur (par exemple lors d'un appel de sous-programme), de manière à pouvoir le rétablir plus tard. Le contenu des registres peut être entré (PUSH) sur la pile, puis ressorti (PULL) plus tard, lorsque le contrôle revient au programme principal. Le pointeur de pile indique simplement au processeur la position du dernier article entré dans la pile, et l'emplacement où il peut sauvegarder ou trouver l'article suivant. Du fait qu'ils renvoient aussi à des adresses mémoire, les pointeurs de pile ont généralement 16 bits de longueur.

Le quatrième registre est très important, bien que sa fonction soit la plupart du temps automatique. C'est le *compteur de programme,* qui doit toujours contenir l'adresse où est stockée l'instruction suivante. Le processeur va à l'emplacement spécifié par le registre, cherche le contenu, interprète son sens et agit sur cette instruction. Normalement, le compteur de programme est incrémenté automatiquement au fur et à mesure des instructions, afin que celles-ci se trouvent en séquence. Si l'on modifie le contenu du compteur de programme (en y stockant une nouvelle valeur, ou en l'additionnant ou la soustrayant de l'ancienne valeur), on changera le cours du programme. Autrement dit, cela a l'effet d'une instruction GOTO, bien que, à ce niveau, on parle plutôt de saut (JMP) s'il s'agit d'une nouvelle adresse, ou d'un branchement (BRA) si l'adresse en cours a été modifiée.

Il y a un cinquième type de registre, quoiqu'il n'opère pas de la même façon que les autres. C'est le registre de *code de condition,* que l'on peut représenter par un ensemble de bits individuels, chacun correspondant à un aspect de l'état du processeur. Par exemple, l'un de ces bits est utilisé pour signaler au processeur que le nombre résultant d'une opération dans l'un des regis-

La préférence du public
Les deux machines les plus populaires basées sur le 6809 sont toutes deux des ordinateurs familiaux : Dragon 32 K et 64 K, et Tandy Color Computer. Il existe aussi une variété de systèmes utilisant le 6809 dans les universités et les grandes écoles. Le Tandy Color Computer et les deux Dragon sont très semblables dans leur structure interne, et bien que Dragon Data ait abandonné le marché de l'informatique, Tandy a accepté de fournir le logiciel et le support technique aux utilisateurs de Dragon.
(Cl. Ian McKinnell.)

tres est nul; ainsi, nous pouvons parcourir un tableau de valeurs en chargeant le nombre total des valeurs dans un registre et en soustrayant un de ce total chaque fois que nous avons affaire à une valeur. Lorsque le total s'annule, le bit de code de condition indique au processeur qu'il n'a plus d'articles dans le tableau à traiter, et qu'il peut passer à une autre instruction. Ce type d'instruction nous permet de faire des choix (instructions IF) et des boucles (FOR...NEXT, WHILE...WEND, REPEAT...UNTIL).

De nombreux processeurs comportent une *page zéro* qui contient normalement les 256 premières positions mémoire (de 0000 à 00FF hex). Les valeurs qui y sont stockées peuvent être adressées par huit bits, ce qui rend les instructions plus courtes et plus rapides à exécuter. Le processeur 6809 généralise ce concept en ayant un registre de *page directe* à 8 bits qui fournit les 8 bits supplémentaires de l'adresse complète lorsqu'on se réfère à la page zéro. En changeant la valeur dans ce registre, on peut positionner la page zéro n'importe où en mémoire, ou même en avoir plusieurs.

Un programme en langage machine consiste en une série d'instructions entremêlées avec des données et des adresses. Certains arrivent à programmer directement en manipulant des valeurs numériques pour toutes ces quantités, mais la plupart trouveraient cela trop difficile. Le langage d'assemblage d'un processeur nous permet d'écrire des programmes en langage machine en utilisant des mnémoniques (pour les instructions) et des labels (pour les adresses et les données), ce qui est bien plus commode. Ainsi, s'il faut charger la donnée à une adresse dans l'accumulateur, on peut écrire :

```
STORE FCB 0
```

pour réserver une place en mémoire que l'on pourra désigner par STORE, dans laquelle nous avons provisoirement mis un zéro. FCB n'est pas une vraie instruction, mais une directive disant au programme qui traduit le langage d'assemblage en langage machine de substituer une adresse particulière chaque fois qu'il rencontre le mot STORE. Ultérieurement dans le programme, quand nous voudrons charger dans l'accumulateur la valeur stockée là, nous pourrons utiliser l'instruction :

```
LDA STORE
```

qui prendra la valeur stockée, quelle qu'elle soit, pour la charger.

Les programmes en langage d'assemblage doivent être traduits avant de pouvoir être exécutés, et c'est la tâche d'un programme appelé *assembleur*. Il ne s'agit pas de programmes compliqués, car il y a presque une relation biunivoque entre les instructions de langage d'assemblage que nous écrivons et leurs équivalents en langage machine. Tout ce qu'il faut faire, ce sont des substitutions, et il ne faut pas oublier quels noms renvoient à quelles valeurs ou adresses.

De nouveaux horizons

Psion a toujours été étroitement associée à Sinclair et lui a fourni différents programmes. Récemment, la compagnie s'est tournée vers les progiciels, et a lancé l'ordinateur de poche Organiser.

Un ensemble intégré
La gamme Xchange de Psion est un ensemble de logiciels de gestion intégrés, créés à partir des quatre programmes créés pour le QL. Quill est un traitement de texte, Archive une base de données, Abacus un tableur utile pour les prévisions financières, et Easel un logiciel de création graphique. Tous quatre peuvent être achetés ensemble ou séparément. Ils sont disponibles pour l'IBM PC et PC XT, pour l'Apricot et le Sirius 1, tandis que des versions sont prévues pour le Macintosh d'Apple et le Rainbow de DEC.

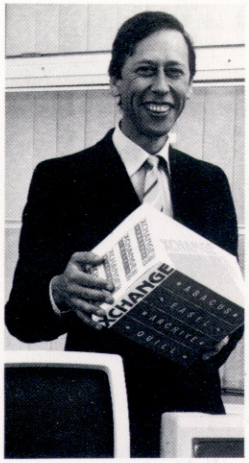
Psion fut fondée en 1981 par le Dr David Potter, assistant à l'Imperial College de Londres. Elle sortit d'abord quatre programmes destinés au ZX81, qui venait de faire son apparition : un simulateur de vol, un jeu de backgammon et deux logiciels « sérieux », Vu-Calc (un tableur) et Vu-File (une base de données). Tous quatre avaient été rédigés par Charles Davies et Colly Myers. Leur qualité était telle que la réputation de la firme fut aussitôt établie. C'est pourquoi, l'année suivante, Sinclair Research décida de lui confier la création d'une cassette qui devait faire la démonstration irréfutable des qualités du Spectrum sur le marché des micro-ordinateurs.

Les ordinateurs de Sir Clive ayant connu un énorme succès en Grande-Bretagne, Psion put ainsi bénéficier d'un très vaste marché. On estime par exemple que le programme de simulation de vol (versions ZX81 et Spectrum confondues) s'est vendu à plus de 500 000 exemplaires, tandis que les ventes de l'ensemble du catalogue de la firme représentent plus de trois millions de cassettes... La création récente des logiciels de la gamme « Xchange » indique que Psion entend s'implanter sur le marché de la gestion informatisée, qui, en Grande-Bretagne seulement, représente plus de deux milliards de francs par an.

La société a toujours cherché à innover : elle fut l'une des premières à généraliser l'emploi de la « compilation croisée », au cours de laquelle un programme destiné à une machine est mis au point sur une autre. « Horizons », le programme de démonstration du Spectrum, a ainsi été rédigé sur un TRS-80. Aujourd'hui, Psion est passé au stade supérieur, et fait usage de deux mini-ordinateurs VAX 750.

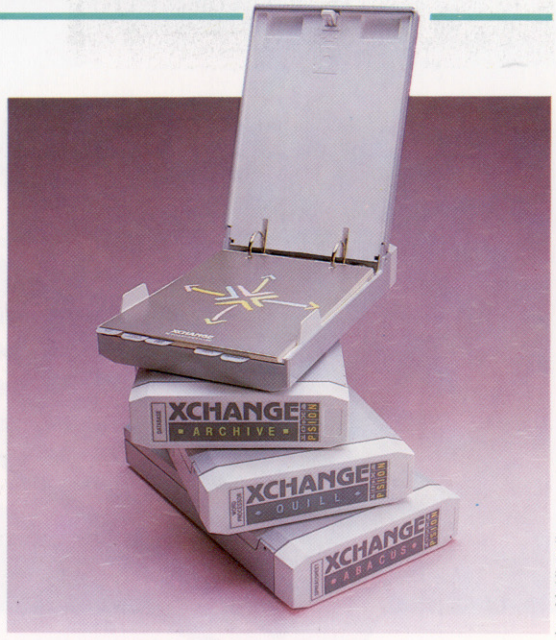
C'est avec eux qu'ont été créés les quatre logiciels qui sont fournis avec le QL : Abacus (un tableur), Archive (une base de données), Easel (un programme graphique) et Quill (un traitement de texte). La firme a même mis sur pied, outre-Manche, un service appelé QLUB, qui garantit une réponse par lettre, sous quarante-huit heures, à toute demande de renseignements de la part des possesseurs de QL.

Le créateur
Universitaire spécialisé dans la physique de l'informatique, qu'il enseigna à l'Université de Californie et à l'Imperial College de Londres, le Dr David Potter fonda Psion (au départ Potter Scientific Investments), et reste son principal actionnaire.



Diversification

Des programmes de gestion destinés à l'IBM PC, à l'Apricot, au Sirius, au Rainbow de DEC et au Macintosh sont actuellement en cours de création. Ils contiennent la gamme « Xchange », et sont très semblables à ceux qui accompagnent le QL. Psion déclare que leur supériorité, par rapport aux progiciels classiques, vient de la facilité avec laquelle on peut transférer les données de l'un à l'autre.



Ian McKinnell

Le lancement récent de l'ordinateur de poche Organiser n'a pas manqué de surprendre beaucoup de gens, pour qui Psion reste avant tout une firme productrice de logiciels. Mais les responsables expliquent : « Psion est, en effet, une compagnie qui édite des programmes. L'Organiser nous a paru avant tout être une très bonne idée de *présentation* et d'*utilisation* de ces programmes. Il n'existait rien de comparable, alors nous avons construit notre propre appareil, mais la mise au point a toujours été conçue en fonction du logiciel. » Comme quoi, même outre-Manche, on n'est jamais mieux servi que par soi-même !

Pour le moment, l'Organiser ne peut encore mettre en œuvre que trois programmes d'application (sciences, mathématiques et finances), ainsi que des blocs-mémoire RAM de 8 et 16 K (les « datapaks »). Mais la firme compte bien en produire d'autres, et a déjà été contactée par des programmeurs indépendants.

Enfin la société cherche à accroître sa part de marché : elle a récemment créé des filiales aux États-Unis et en Afrique du Sud, tout en signant des contrats pour la distribution de ses produits dans toute l'Europe. Sinclair Research vient d'aborder le marché d'Europe de l'Est en exportant quatre cents ZX81 vers la Tchécoslovaquie, et Psion, qui prévoit le lancement de versions en langues étrangères de certains de ses titres, aura sans doute son mot à dire dans ce développement international du leader britannique.

Page manquante
(publicité)

Page manquante
(publicité)